

# **UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE QUITO – CAMPUS SUR**

**CARRERA DE INGENIERÍA DE SISTEMAS**

**MENCIÓN ROBÓTICA E INTELIGENCIA ARTIFICIAL**

**DISEÑO Y CONSTRUCCIÓN DE UN API (INTERFAZ DE PROGRAMACIÓN DE  
APLICACIONES) PARA LA PLATAFORMA .NET QUE PERMITA CONTROLAR EL  
BRAZO ROBÓTICO MITSUBISHI RV-2AJ A TRAVÉS DEL PUERTO SERIAL.**

**TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS**

**FREIRE MUÑOZ ANDRÉS SANTIAGO**

**MARTÍNEZ CORAL MARSAL PATRICIO**

**DIRECTOR**

**ING. RODRIGO TUFIÑO**

**QUITO, ENERO 2012**

## DECLARACIÓN

Nosotros, Andrés Santiago Freire Muñoz y Marsal Patricio Martínez Coral, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

---

Andrés S. Freire Muñoz

---

Marsal P. Martínez Coral

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Andrés Santiago Freire Muñoz y Marsal Patricio Martínez Coral, bajo mi dirección.

---

Ing. Rodrigo Tufiño

## **DEDICATORIA**

Dedicado a todas las personas que han estado siempre respaldándome y dándome ánimos para conseguir mis metas y mis objetivos. A mis amigos por estar a mi lado en los momentos difíciles y a los obstáculos del camino por recordarme que las cosas buenas merecen esfuerzo.

Pero muy especialmente a mi madre, gracias a quien soy lo que soy ahora y quien me enseñó a nunca claudicar ante nada ni ante nadie.

Patricio Martínez

A Dios, quien me permitió conservar la fuerza y la salud en los momentos más difíciles, a mis padres y mi tía quienes son pilares de mi desarrollo intelectual y espiritual, a quienes dejaron de ser profesores y se convirtieron en maestros, que supieron llevar con temple y decisión su cátedra, para poder sustentar las bases de mi conocimiento en mi especialidad, a mis amigos, con quienes pasamos por obstáculos y salimos adelante, y por último a quienes moldearon mi razón y el deseo de salir siempre adelante, para hacer de cada paso el más importante y fundamental en mi vida, mis abuelitos (en paz descansen).

Andrés Freire

## **AGRADECIMIENTOS**

La realización de este proyecto de tesis ha sido gracias a la colaboración de las siguientes personas:

A nuestros padres y familia por darnos el apoyo incondicional en todo proyecto que hemos emprendido.

A nuestro director de tesis, el Ing. Rodrigo Tufiño, por haber sido más que un maestro, un amigo.

Al Tglo. José Zapata, quien nos demostró que nuestro proyecto era posible.

A nuestros amigos por ser nuestro aliento cuando nos faltaba aire y a nuestros rivales por mostrarnos que siempre habrá alguien trabajando igual o más que nosotros y exigirnos siempre mejorar y jamás permitirnos conformarnos.

## CONTENIDO

RESUMEN .....	5
PRESENTACIÓN.....	6
1. INTRODUCCIÓN .....	7
1.1 MODELADO DEL ENTORNO .....	8
1.2 CONTROL DEL MOVIMIENTO DEL ROBOT .....	8
1.3 PROGRAMACIÓN MANUAL .....	9
1.4 OBJETIVOS .....	11
1.5 ALCANCE DEL PROYECTO .....	12
2. ESTADO DEL ARTE .....	13
2.1 BRAZO ROBÓTICO .....	20
2.1.1 CONTROLADOR .....	20
2.1.2 GRADOS DE LIBERTAD (GDL) .....	21
2.1.3 ESPACIO DE TRABAJO .....	23
2.1.4 EXACTITUD DE LOS MOVIMIENTOS.....	24
2.1.5 VOLUMEN DE CARGA .....	25
2.1.6 RAPIDEZ.....	26
2.1.7 ACTUADORES.....	26
2.1.8 PROGRAMABILIDAD .....	31
2.2 TIPOS DE BRAZOS ROBÓTICOS .....	32
2.2.1 SCARA (Selective Compliance Assembly Robot Arm).....	32
2.2.2 CARTESIANOS .....	33
2.2.3 CILÍNDRICOS .....	34
2.2.4 ESFÉRICO / POLAR .....	35
2.2.5 ARTICULACIÓN TIPO PUMA.....	35
2.3 USO DE LOS ROBOTS INDUSTRIALES .....	36
2.3.1 AUTOMATIZACIÓN EN LA INDUSTRIA DEL AUTOMÓVIL .....	41
2.3.2 AUTOMATIZACIÓN EN EL ÁMBITO DE LOS PROVEEDORES DE LA INDUSTRIA DEL AUTOMÓVIL. ....	43
2.3.3 AUTOMATIZACIÓN EN LA INDUSTRIA ALIMENTICIA.....	44

2.3.4	AUTOMATIZACIÓN EN LA INDUSTRIA DE LA MADERA .....	46
2.3.5	AUTOMATIZACIÓN EN LA INDUSTRIA DEL PLÁSTICO.....	48
2.3.6	ROBOTS EN LA INDUSTRIA DE LA FUNDICIÓN.....	49
2.3.7	ROBOTS QUE CIERRAN HERMÉTICAMENTE .....	53
2.4	UTILIZACIÓN DE BRAZOS ROBÓTICOS EN EL ECUADOR.....	60
2.5	ROBÓTICA EN EL ÁMBITO EDUCATIVO .....	61
3	ANÁLISIS Y DISEÑO .....	65
3.1	DESCRIPCIÓN DEL HARDWARE Y SOFTWARE.....	65
3.1.1	HARDWARE DE CONTROL .....	65
3.1.2	TEACH .....	70
3.1.3	MANIPULADOR.....	72
3.1.4	EFECTOR FINAL O GRIPPER.....	72
3.1.5	RANGO DE OPERACIÓN .....	74
3.1.6	ESPECIFICACIONES TÉCNICAS.....	76
3.1.7	ENTORNO DE TRABAJO .....	78
3.2	DESARROLLO DEL SOFTWARE .....	133
3.2.1	CASOS DE USO .....	134
3.2.2	DIAGRAMAS DE CLASES.....	141
3.2.3	DIAGRAMAS DE SECUENCIA .....	145
3.3	MODELADO DE INTERFAZ .....	149
3.3.1	PANTALLA PRINCIPAL .....	149
3.3.2	PANTALLA MOVIMIENTO JOINT .....	150
3.3.3	PANTALLA MOVIMIENTO POR POSICIONES ALMACENADAS.....	151
3.3.4	PANTALLA CONFIGURACIONES .....	152
3.4	PROGRAMACIÓN.....	153
3.4.1	LENGUAJE DE PROGRAMACIÓN MELFA BASIC IV .....	153
3.4.2	PROTOCOLO RS-232 .....	156
3.4.3	DESAFÍOS DEL MERCADO .....	157
4	CONSTRUCCIÓN E IMPLEMENTACIÓN .....	158
4.1	PROGRAMACIÓN DEL SOFTWARE.....	158
4.2	MÉTODOS PARA MOVER EL BRAZO .....	159

4.2.1	CONDICIONES DE INICIO .....	159
4.2.2	MOVER EN XYZ .....	159
4.2.3	MOVER POR ARTICULACIÓN .....	160
4.3	LEER LOS PROGRAMAS ALMACENADOS EN LA MEMORIA DE LA CONTROLADORA .....	161
4.4	CLASE PARA COMUNICACIÓN E INSTRUCCIONES DE MOVIMIENTO .....	162
4.4.1	MÉTODO PARA DEFINICIÓN DE PARÁMETROS DE COMUNICACIÓN.....	162
4.4.2	MÉTODO PARA ENVÍO DE DATOS A LA CONTROLADORA.....	162
4.4.3	MÉTODO PARA ENVÍO DE COMANDOS A LA CONTROLADORA.....	163
4.4.4	MÉTODOS PARA ACTIVACIÓN Y DESACTIVACIÓN DE SERVOS.....	163
4.4.5	MÉTODOS PARA ABRIR Y CERRAR GRIPPER .....	164
4.4.6	MÉTODOS PARA DESPLAZARSE A UNA POSICIÓN DETERMINADA .....	164
4.5	PROGRAMACIÓN DEL ROBOT .....	165
4.6	RESUMEN DE PROTOCOLOS Y COMANDOS Y CONTENIDOS.....	169
4.7	INTERFAZ FINAL DEL SOFTWARE.....	171
4.7.1	PANTALLA PRINCIPAL .....	171
4.7.2	PANTALLA DE CONFIGURACIÓN.....	172
4.7.3	PANTALLA DE MOVIMIENTO EN MODO XYZ.....	172
4.7.4	PANTALLA DE MOVIMIENTO POR ARTICULACIONES .....	173
4.7.5	PANTALLA DE MOVIMIENTO POR POSICIONES.....	174
4.7.6	TEACH .....	175
4.7.7	EJEMPLO.....	177
4.8	PRUEBAS Y AJUSTES .....	177
4.9	CONCLUSIONES Y RECOMENDACIONES.....	181
4.9.1	CONCLUSIONES .....	181
4.9.2	RECOMENDACIONES .....	183
5	ANEXOS .....	185
5.1	MÉTODOS CLASE CLSRV-2AJ .....	185
5.2	MANUAL DE USUARIO.....	187
5.2.1	PANTALLA PRINCIPAL DEL PROGRAMA.....	187
5.2.2	PANTALLA DE CONFIGURACIÓN.....	190
5.2.3	PANTALLA DE MOVIMIENTO EN MODO XYZ.....	191



5.2.4	PANTALLA DE MOVIMIENTO POR ARTICULACIÓN.....	194
5.2.5	PANTALLA DE MOVIMIENTO POR POSICIONES.....	196
REFERENCIAS WEB .....		200

## **RESUMEN**

El presente trabajo es un proyecto que permite integrar de manera fácil y potente, el hardware del Robot RV-2AJ Mitsubishi del laboratorio de MPS de la UPS con software compatible a la plataforma .NET. Esta integración permite ampliar las capacidades del brazo robot, sin que esto signifique un aumento en los costos de desarrollo para la Universidad o para el estudiante. Este aumento de capacidades se puede dar a través de la utilización de módulos de visión e inteligencia artificial.

El proyecto utiliza el protocolo de comunicación serial de Mitsubishi y el lenguaje MELFA BASIC IV, para generar una librería que puede ser integrada con otras aplicaciones desarrolladas sobre diferentes lenguajes soportados en la plataforma Microsoft .Net.

El siguiente trabajo describe el diseño y construcción del software que compone el API, su estructura y su funcionamiento. Además se detalla el análisis, diseño e implementación del software que controla el robot.

La funcionalidad del API comprende varios puntos como por ejemplo, la manipulación del brazo robótico a través de la controladora vía serial, lo cual permite tener una estructura de control más fácil de utilizar para la generación de movimientos, carga de programas que están grabados en la controladora, y de esta forma tener un control más adaptable tanto para el entorno de la estación como para la concepción espacial del usuario al brazo robótico.

## **PRESENTACIÓN**

En los países desarrollados la robótica es el eje fundamental de las industrias de producción, mientras que aquí en el Ecuador está empezando a crecer.

Hacer que los estudiantes universitarios estén cada vez más relacionados con los avances que tiene la robótica, permitirá que en un futuro nuestro país cuente con profesionales más capacitados y emprendedores, quienes serán fuentes de nuevas ideas y soluciones, y generen el cambio que nuestra industria requiere, para colocarnos en el mapa como un referente de desarrollo a nivel continental y mundial.

Uno de los inconvenientes que tiene la robótica para la Universidad Ecuatoriana, es que estamos sujetos a desarrolladores del extranjero, ya que no cuenta con la tecnología suficiente para generar este tipo de herramientas, como son los brazos robóticos que se producen en Japón, Alemania, Estados Unidos.

Por este motivo el presente trabajo, busca eliminar en parte esa dependencia, al realizar una herramienta de software libre, que permite la transferencia de tecnología, al darle una utilización más adecuada a nuestras necesidades a una herramienta de alta tecnología desarrollada en un país de primer mundo, al pasar las aplicaciones robóticas simuladas a aplicaciones robóticas prácticas, sin que una licencia o software ajeno a la universidad limite su utilización.

Estamos seguros que el presente trabajo será una herramienta muy valiosa, para los estudiantes y los profesores, al darles la herramienta para aplicar sus conocimientos de robótica con hardware robótico de alta calidad, abriendo un panorama totalmente nuevo para aplicaciones que al momento no son posibles de realizar.

## 1. INTRODUCCIÓN

El presente proyecto de titulación tiene por objeto aprovechar el hardware del brazo robótico RV2AJ, para poder utilizar todas sus funcionalidades y otras adicionales como control remoto, programación mediante el uso de redes neuronales para el aprendizaje de movimientos, visión artificial para el reconocimiento del entorno, mediante el uso del API que se va a entregar al final del desarrollo. Para entender de mejor manera el presente proyecto se presenta la siguiente explicación.

Como en cualquier lenguaje de programación convencional, es de gran importancia tener una comprensión de las estructuras de programación, para obtener un incremento considerable de calidad en la producción al igual que contar con un entorno de desarrollo adecuado a las necesidades.

Las operaciones para la programación de un robot, es en principio complejo, en función de que generalmente existe una interacción cambiante con relación al entorno, convirtiéndose en un asunto dinámico.

Esta es la principal causa que lleva a que la mayoría de los métodos de clasificación de robots sean de tipo manual, consiguiendo ejecutar un proceso paso a paso de lo programado en cada momento.

Es importante, que un software para robots denote una correcta capacidad de depuración y ejecución controlada, tomando en consideración la interacción en tiempo real que existe entre los distintos equipos de la célula con el controlador del robot. Es recomendable la realización de una buena acción de monitoreo continuo en relación a la ejecución del desarrollo del programa.

Es de destacar que, en coincidencia con otras tecnologías industriales, la robótica educativa debe utilizar un entorno de sistemas operativos de amplia difusión como Windows o Linux, por la facilidad que esto le supone al usuario al encontrarse familiarizado con estos sistemas.

Otras prestaciones que algunos robots incluyen, bien de serie o de manera opcional, en sus entornos de desarrollo son: la disponibilidad de herramientas para la programación fuera de línea (sin necesidad de disponer del robot), el uso de simuladores de la célula robotizada con posibilidad de generar el programa directamente para el robot, o la existencia de herramientas para el desarrollo de la interface humano-máquina.

## **1.1 MODELADO DEL ENTORNO**

Se puede definir como la relación que posee el robot con los objetos que lo rodea. Generalmente, este modelo está limitado a tipos y relaciones de figuras geométricas tomando en cuenta su posición, orientación y ocasionalmente a su forma, dimensiones, peso, etc.

Para precisar la colocación y ubicación de los objetos del entorno, lo más habitual es designar a cada objeto de manera individual en un medio de referencia, de tal forma que la colocación y ubicación de este medio referido sea el sistema base.

## **1.2 CONTROL DEL MOVIMIENTO DEL ROBOT**

Innegablemente, un método de programación para robots debe tener la posibilidad de puntualizar el movimiento del robot. Conjuntamente con el lugar de llegada, es necesario puntualizar el tipo de recorrido espacial que debe ser ejecutado, la velocidad de configuración y la precisión con que se debe llegar al sitio destino.

De ser necesario se debe indicar si el movimiento debe desarrollarse en cualquier caso o debe estar limitado por algún tipo de sensor.

Un robot usualmente no trabaja como un elemento separado, sino que es parte de un proceso de manufactura que se conforma de algunos factores como de la llamada célula de trabajo robotizada.

### **1.3 PROGRAMACIÓN MANUAL**

La programación manual o por aprendizaje radica en permitir trabajar al robot, de manera manual una vez que se hayan tomado los registros y las configuraciones adoptadas por la práctica, para su posterior recreación de forma automática.

Para configurar al robot por los recorridos o puntos considerados se manejan algunas soluciones. Cuando los actuadores del robot estén deshabilitados y así el programador debe poder acoplar la energía suficiente para generar los desplazamientos del robot, entonces se tiene como resultado un control pasivo por parte del usuario. De esta manera, el programador obtiene el límite extremo del robot y así podrá desplazarlo hasta los sitios deseados tomando las trayectorias más apropiadas. El mecanismo de control del robot registrará, de forma independiente, el registro de los sensores de estado de las articulaciones en todos los puntos recorridos.

La complejidad física de desplazar la estructura del robot se resuelve con la utilización del guiado pasivo dependiendo de la clase de manipulador.

En este caso, el registro de las configuraciones que adopta el robot es continuo, es decir, la unidad de control muestrea y guarda con cierta periodicidad elevada, las perspectivas de las articulaciones, el recorrido seguido por el robot y su rapidez queda detallada en el historial de configuraciones.

Frente a estos sistemas pasivos, otra posibilidad permite emplear el propio módulo de accionamiento del robot, manejado desde un panel de control o un dispositivo de mando para el control manual, llamado Teach, ubicado en un indicador de programación móvil para que sean los eventos los delegados de conducir las articulaciones.

Lo usual es que el dispositivo de control exclusivamente registre aquellas configuraciones del robot, las cuales son descritas por el programador.

El proceso de programación se establece en mover las articulaciones mediante la ventana de programación o las instrucciones que vengan del dispositivo o interfaz de programación, al robot hasta una posición establecida para luego grabar esta configuración. De esta forma, el desplazamiento que ha usado el robot para alcanzar la configuración designada, precisa ser añadida a los datos que delimitan el recorrido del robot, desde la configuración anterior hasta la nueva.

Adicionalmente a través del panel de programación se pueden incluir instrucciones para el control del flujo del programa (saltos, repetición de movimientos), atención a entradas/salidas binarias, etc.

Los sistemas de programación manualmente controlados son muy útiles e incluso necesarios en muchas ocasiones. Conjuntamente presentan ventajas, como el no requerir de coordenadas de los elementos principales del entorno de trabajo, no se producen errores de posicionamiento por una inadecuada graduación del robot o su entorno.

La mayoría de los robots en la actualidad poseen una plataforma estática, y esta manufactura está basada en trabajos de producción industriales como montaje, acoplamiento de soldadura, paletizado, etc. Además, se encuentran otra variedad de aplicaciones que han evolucionado en función de su estructura a través de los años. Algunos de estos robots son utilizados en aplicaciones no industriales, pueden ser usados en robots espaciales, brazos para desplazamientos y control de satélites, móviles de investigación lunar, usados en la construcción, para uso submarino y subterráneas, localización y manutención de cables marítimos, limpieza y observación de conductos y desagües bajo tierra, reconocimiento, control de procedimientos de enfriamiento nuclear, robots marciales para desactivación o detonación de explosivos, celadores experimentales usados para inspeccionar perímetros críticos, atención y salvamento (bomberos, patrullas para controlar accesos en fábricas, aéreos); estudios y cuidados médicos (cirujanos, prótesis biónicas, estructuras de ayuda para personas con capacidades especiales); para uso agrícola (sembrado y cosechas, para rasurar ovejas) entre otros.

## **1.4 OBJETIVOS**

### **Objetivo General**

Construir un API para la plataforma .NET que permita el control del brazo robótico Mitsubishi RV-2AJ a través del puerto serial.

### **Objetivos Específicos**

- Estudiar y entender el lenguaje de programación MELFA BASIC IV.
- Realizar una revisión bibliográfica de las tecnologías a utilizar e investigar el funcionamiento del brazo robótico.
- Analizar los requerimientos del hardware y software para controlar el brazo robótico a través de la plataforma .NET.
- Investigar la configuración del protocolo de comunicación serial para el brazo MITSUBISHI RV-2AJ.
- Investigar la sintaxis de comunicación con el brazo robótico RV-2AJ vía serial.
- Desarrollar una DLL que permita el control del brazo y la integración con la plataforma .NET.
- Desarrollar una aplicación de ejemplo que permita demostrar el funcionamiento del API



## 1.5 ALCANCE DEL PROYECTO

El presente proyecto de titulación contempla la creación de un API a través de un DLL, desarrolladas en C#, que implementen funciones del lenguaje MELFA BASIC IV, que permitan conectarse vía serial con el brazo robótico MITSUBISHI RV-2AJ, para realizar las siguientes tareas:

- encendido y apagado de servos
- Reseteo de la alarma de error
- abrir y cerrar pinza neumática
- Movimiento del brazo robótico en modo XYZ
- Movimiento del brazo robótico en modo JOINT.
- Desplazamiento entre posiciones del programa seleccionado

Utilizaremos las configuraciones de fábrica para la comunicación serial con el brazo, por lo que no desarrollaremos protocolos nuevos de comunicación.



Figura 1.1 Robot RV-2AJ

Fuente: Web1

## **2. ESTADO DEL ARTE**

Dentro de la categoría de Robots, se destacan los Brazos Robóticos ya que a nivel tecnológico son los proyectos más utilizados debido a sus funcionabilidades y practicidad en sus procesos de manufactura, a través de los años se han incrementado a nivel mundial principalmente debido al nivel de trabajo y tiempo de producción que alcanzan permitiendo al ser humano preocuparse de labores con más raciocinio y de gestionamiento de la producción o proyección de ventas, debido a estas razones se tiene una amplia gama de categorías y funcionalidades a nivel mundial, los cuales se presentarán más adelante.

### **Antecedentes**

La disposición de los robots preliminares revelaba a las nombradas estructuras esféricas y cartesianas, para producción principalmente basado en la manipulación.

Un logro significativo en el esquema de robots es acoplamiento directo, perfeccionado en la Universidad Carnegie Mellon en 1981. Los motores se ajustan directamente a las uniones sin el uso de moto reductores, lo que admite desplazamientos más veloces y exactos.

Además en los años setenta los robots empezaron a ser utilizados de manera útil en otros ambientes, innovando su uso especialmente en el área de tecnologías de control vía remota.

Se pueden encontrar otras definiciones originarias del avance tecnológico que se han destacado sobre las limitaciones incriminadas por el sector industrial.

La claridad con la que nuestra sociedad trata al robot es más interesante cuando se confronta con el extenso desconocimiento que se puede tener de otros autómatas o aparatos, inclusive si estos son de mayor antigüedad o beneficio teniendo como ejemplo el osciloscopio o las generaciones próximas de los robots: los sistemas de control numérico.

En los últimos años, una nueva curiosidad ha aparecido en Japón en función de los robots de forma humana.

La expresión robot, probablemente, se hubiera consumido si no cobrase fuerza en los escritores de libros de ciencia ficción, la mayoría de los cuales profundizaron el mensaje, del trabajo de Capek: “la dominación de la especie humana por seres hechos a su propia imagen”<sup>1</sup>. De esta forma, en 1926, Thea von Harbou redacta *Metrópolis*, fantasía contiguamente llevada a la pantalla grande, donde un grupo obrero de una comunidad super-industrializada es manejada a través de un cabecilla androide con el nombre de María.

Pero quien llenó la mente de los aficionados a los libros de ciencia ficción, fue el novelista americano de descendencia rusa Isaac Asimov (1920-1992) quien fue el mayor promotor del término robot.

En la década de los años '40 anunció en la revista *Galaxy Science Fiction* una obra llamada «The Caves of Steel» fue ahí donde por primera vez formuló sus tres leyes de la robótica<sup>2</sup>.

1. “Un robot no puede perjudicar a un ser humano, ni con su inacción permitir que un ser humano sufra daño.
2. Un robot ha de obedecer las órdenes recibidas de un ser humano, excepto si tales órdenes entran en conflicto con la primera ley.
3. Un robot debe proteger su propia existencia mientras tal protección no entre en conflicto con la primera o segunda ley”.

Los mecanismos iniciales que reconocen al que hoy se denomina con el nombre de robot, no acogieron en un principio esta definición. Luego de los autómatas

---

<sup>1</sup> BARRIENTOS, Antonio y otros, Fundamentos de la Robótica

<sup>2</sup> En la novela *Robots e Imperio*, publicada en 1985, Asimov incorporó una cuarta ley, conocida como ley cero: un robot no puede lastimar a la humanidad o, por falta de acción, permitir que la humanidad sufra daño. Esta ley, de mayor prioridad que la primera (que debe ser modificada en tal sentido), antepone el bien comunitario al individual.

inicialmente descritos en los párrafos anteriores, en su mayoría de aspecto humano, los antecesores más directamente relacionados con los robots debieron ser los teleoperados

El motor del ser humano es la necesidad, por lo que siempre ha estado buscando la manera de desarrollar ciencia y tecnología. Uno de estos desarrollos ha sido la robótica, que para nuestra época, es un avance tan dramático como han sido las computadoras desde mediados del siglo XX. La robótica aporta al desarrollo de muchos campos de nuestra vida como son la industria, la medicina, la educación, la exploración entre otros, lo que permite avanzar de una manera mucho más veloz que hace 20 años.

Todos estos avances significan un gran beneficio para las sociedades pero también significan un gran costo en inversión que no todos los países están en capacidades de afrontar, por lo que se opta, por comprar a países que ya hayan desarrollado estas tecnologías e invertir en nuevas aplicaciones en lugar de tratar de desarrollar todo desde cero, aprovechando el conocimiento de otras personas en beneficio propio; a esto se lo llama transferencia de tecnología.

La transferencia tecnológica es una forma de propagación de conocimiento, que se da, entre países desarrollados y países en vías de desarrollo. Lo que impulsa el desarrollo y el crecimiento, de los países en vías de desarrollo.

En 1947 R. C. Goertz del Argonne National Laboratory creó, con la meta de manipular dispositivos radioactivos sin peligro para el usuario calificado, el sistema inicial de control remoto. Esto radicaba en un módulo mecánico de configuración maestro - esclavo. El robot maestro, ubicado en el perímetro seguro, era manejado solamente por el usuario, y el esclavo, ubicado directamente con las sustancias radioactivas y unidas estructuralmente al maestro, que a su vez seguía de manera exacta los desplazamientos de éste. El usuario también podía detectar a través de un cristal de seguridad el movimiento consecuente de sus operaciones.

Luego en 1954, Goertz hizo uso de los procesos electrónicos y de servo control reemplazando la operación mecánica por otra electrónica y de esta manera tuvo un desarrollo del primer prototipo de teleoperación mediante servo control bidireccional, junto con la manufacturación nuclear, en el recorrido de los años sesenta la explotación submarina empezó a relacionarse con el uso de operadores remotos. A este beneficio se añadió la carrera espacial en la década de los setenta.

El reemplazo del usuario por un sistema que interviniese en los desplazamientos del robot, permitió el paso a la definición de robot, instigado por los requerimientos de automatización de las cadenas de manufactura.

Para una gran mayoría en la actualidad de la humanidad un robot se define como una estructura de mecanismos estructurales y electrónica, idóneo para trabajar de forma permanente, elaborando recursos de consumo masivo. Los robots proponen renovación y avance en el sector de la automatización, apareciendo en períodos anteriores, actuales y ciertamente futuros, como una de las expectativas del avance tecnológico.

Los manipuladores industriales, nacidos a inicios de los sesenta, consiguieron sortear grandes dificultades, tanto técnicas como socio económicas y lograron su primera madurez en la los años noventa, afianzándose de esta manera su necesidad de producción en serie y obteniendo una aprobación y afirmación robusta en la industria.

Con el creciente uso del robot industrial, los técnicos e investigadores enfocados en la robótica decidieron tratar de dar un paso fuera del sector productivo, otorgándoles de la capacidad de conducirse en otros ambientes, poco o nada estructurados, que sean capaces de colaborar con personas no especializadas en su uso de manera inteligente.

Tomando en cuenta la gran problemática técnica de este objetivo, los esfuerzos ejecutados y la adaptación de los adelantos obtenidos en los métodos que sirven de ayuda a la robótica, ha dotado un lugar a que en la naciente década del siglo XXI se

llegue a un nuevo renacimiento de la robótica, desarrollándose en el medio industrial y expandiéndose en dos diferentes sectores: la robótica de servicios cooperativos y la robótica social.

En los servicios cooperativos se han obtenido desarrollos netamente operativos, de tal forma como se puede comprobar en robots para cirugía, mantenimiento de vidrios de edificaciones, o para seguridad.

La robótica social va a ser, una de las más grandes revoluciones de la robótica de la siguiente década, aportando de manera notoria, al acoplamiento de un nuevo modo de vida de la sociedad. Este apartado no es el producto de una visión fuera de lugar de la gente a nivel global. Bill Gates, fundador y propietario de Microsoft, afirma en su artículo “A Robot in Every Home”<sup>3</sup> (un robot en cada hogar), mediante el surgimiento de la robótica está desarrollándose de una manera equivalente como se innovó en la producción de los computadores hace 35 años, que ha conseguido situar uno, o varios, ordenadores personales en la mayoría de los hogares.

El constante desarrollo que los adelantos en robótica y tecnologías con fin común poseen en la actualidad, causa que la definición de robot deba ser examinada y perfeccionada con reiteración. Por esta razón, es complejo delimitar lo que debe ser comprendido como un robot y luego poder precisar esta palabra con la suficiente generalidad como para envolver el extenso conjunto de mecanismos que son, hoy en día, contruidos. De esta forma, puede resultar oportuno examinar las definiciones para este término conocido como robótica:

“Máquina operada automáticamente que sustituye el esfuerzo de los humanos, aunque no tiene por qué tener apariencia humana o desarrollar sus actividades a la manera de los humanos.”<sup>4</sup>

---

<sup>3</sup>Publicado en la edición de enero del 2007 de la revista Scientific American

<sup>4</sup> Enciclopedia Británica.

“Máquina que se asemeja a los humanos y desarrolla como ellos tareas complejas como andar o hablar Un dispositivo que desarrolla de manera automática tareas complicadas, a menudo de manera repetitiva Un mecanismo guiado por control automático.”<sup>5</sup>

“Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas sólo a las personas.”<sup>6</sup>

En la actualidad dichas definiciones pueden resultar insuficientes para generalizar la totalidad de funciones que involucran a los sistemas robóticos.

Debido a esto es común incluir un calificativo a la palabra robot, que conlleve el adaptar con mejor referencia sus funciones o su entorno de concentración como inteligente o adaptable.

No fue muy común esta gran variedad de sistemas robóticos en décadas anteriores. A finales de 1980 y dejando de lado la ciencia ficción, los primeros robots en la producción, excluyendo el área de los laboratorios, fueron aquéllos utilizados en la manufactura de servicios en talleres y entornos de fabricación de enlatados.

Dichos robots, acoplados en la manufactura, se encuentran hoy en día sólidamente reconocidos, con procesos bien definidos y una amplia gama de mercado.

El desempeño que esta clase de robot ejecuta es principalmente, la de paletizado<sup>7</sup> de piezas o herramientas en un ambiente industrial, por lo que se conoce como robot industrial de paletizado.

En función de esto se ha podido establecer una precisa clasificación según diferentes criterios, como se verá en los siguientes párrafos.

---

<sup>5</sup> Diccionario Merrian Webster.

<sup>6</sup> Diccionario de la Real Academia Española.

<sup>7</sup> armazón de madera, plástico u otros materiales empleado en el movimiento de carga ya que facilita el levantamiento y manejo con pequeñas grúas

## Robot Industrial Operador

Robot (RIA): Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas.

Siendo esta la primera definición, ajustada y delimitada, sirvió de guía para los siguientes postulados que han ido evolucionando para dar continuidad al actual, designado por la (ISO) Asociación Internacional de Estándares.

Robot manipulador industrial (ISO): Manipulador de 3 o más ejes, con control automático, reprogramable, multiplicación, móvil o no, destinado a ser utilizado en aplicaciones de automatización industrial. Incluye al manipulador (sistema mecánico y accionado) y al sistema de control (software y hardware de control y potencia).

La definición ISO establece que el multiarticulado posea mínimo 3 grados de libertad poniendo a un lado de esta definición a ciertos dispositivos estructuralmente menos elaborados de desplazamientos limitados.

La posibilidad de ser reprogramado sin adaptaciones estructurales para obtener una amplia gama de posiciones se necesita de servo control que generará los desplazamientos de manera que se suspendan mientras la posición final sea igual con la requerida.

## Definición de otros tipos de robots

Para determinar a los robots donde la actividad sea fabricar segmentos o bienes a través de técnicas industriales de aquéllos cuya tarea sea diferente se tendrá que:

Robots de servicio (IFR): Un robot que opera de manera semi o totalmente autónoma para realizar servicios útiles a los humanos y equipos, excluidas las operaciones de manufactura.

Estos articulados o manipuladores industriales, pueden ingresar en la categoría de servicio mientras sus procesos de producción no sea la manufactura. Estos pueden



estar provistos o no de un brazo robótico en relación a los industriales. Generalmente este tipo de robots pueden ser comúnmente móviles. En la mayoría de los casos, estos robots están constituidos de un almacén móvil con diferentes tipos de brazos, que se controlan de igual forma que los industriales.

## **2.1 BRAZO ROBÓTICO**

Un brazo robótico es un robot con forma de un brazo humano, diseñado para cumplir funciones repetitivas en las cuales un brazo normal desempeñaría con cierta dificultad.

### **2.1.1 CONTROLADOR**

Registra cada uno de los movimientos del manipulador, las operaciones, cálculos y procesamiento de la codificación. El controlador recibe y envía señales a otros procesos (por medio de señales de entrada/salida) y almacena los programas.

Se registran varios grados de control que están en función del tipo de cuantificaciones que se regulan a partir de una variable o petición del ordenador, a continuación algunos tipos de controladores:

- De posición: el controlador ejecuta las acciones cuando el control determina la posición final del elemento.
- cinemático: el control se determina considerando la posición y la velocidad.
- dinámico: regula la velocidad, la posición y además controla las características solicitadas del manipulador y los elementos vinculados a él.
- adaptativo: determina todas las medidas anteriores y además, permite controlar la variación de las propiedades del manipulador al cambiar de posición.

La mayoría de los robots que en la actualidad se producen con estándares industriales se controlan a través de un recurso llamado, bucle cerrado, es decir, generando un bucle de realimentación.

Los registros que son tomados desde el teach <sup>8</sup> se correlacionan con la del parámetro inicial deseado y se genera en función del error recibido de tal forma que el lugar real del brazo coincide con el que se había establecido inicialmente.

### **2.1.2 GRADOS DE LIBERTAD (GDL)**

Considerando cada uno de los movimientos autónomos (torques y desplazamientos) que determina una articulación en función de la anterior. Esto determina las variables que se necesitan para establecer la posición y la dirección del mecanismo final del manipulador.

El valor de grados de libertad del robot se determina por el conteo de los GDL de las juntas que lo conforman.

Debido a que las articulaciones usadas suelen ser exclusivamente de traslación y prismáticas, con solo un grado de libertad cada una, el número de GDL del robot se determina en relación al número total de articulaciones.

Para ubicar y dar dirección a un cuerpo en cualquier lugar en el entorno de trabajo son requeridas seis variables, tres para relacionar la posición y tres para la dirección, si se desea que un robot posicione y oriente su efector final en cualquier dirección en el espacio, precisará de seis grados de libertad o más.

---

<sup>8</sup> Dispositivo de control y programación manual del Brazo RV-2AJ



Figura 2.1.2.1- Brazo robótico tipo puma

Fuente: Web2

Se presenta el esbozo de un robot de estructura completa con 6 GDL; de los cuales tres de ellos coordinan el lugar del efector final en el entorno de trabajo y los otros 3, la ubicación del robot.

Si se agregan más articulaciones se determinará un aumento en el alcance de la orientación del dispositivo terminal. A pesar que la gran parte de los requerimientos fabriles, usan comúnmente de 6 GDL, principalmente se requiere en las estaciones de soldadura, mecanizado y paletización, para otras aplicaciones más complejas se necesita un número mayor no solo en GDL sino también de módulos completos de trabajo, como es el caso de tareas de montaje. Si se realizan tareas de trabajo en un ambiente con obstáculos, se debe proporcionar al robot mayor alcance y esto genera más grados de libertad los cuales le permitirán tener mayor alcance y orientaciones de su extremo a las que como consecuencia podrá atravesar obstáculos.

Tomando en cuenta los movimientos del multiarticulado y de la muñeca, se puede establecer la cantidad de grados de libertad que posee un robot.

### 2.1.3 ESPACIO DE TRABAJO

Las proporciones de las articulaciones del robot, junto a los grados de libertad, precisan el entorno de trabajo del robot, las cuales definen las propiedades principales en las etapas de clasificación y creación del patrón apropiado.

El espacio de trabajo se puede establecer en diferentes superficies, determinados por el alcance característico del efector final, es diferente a la que permite colocar verticalmente o con un determinado sentido de movimiento.



Figura 2.1.3.1 Brazo robótico UPS Sur <sup>9</sup>

Igualmente queda establecida la zona de trabajo por los sentidos de giro y traslado que militan en las articulaciones.

La cantidad de trabajo de un robot representa exclusivamente al lugar por el cual puede trasladar la parte final de su muñeca. Para establecer la cantidad de producción que se genera o se va a establecer, no se considera la articulación

---

<sup>9</sup> Fuente Laboratorios Universidad Politécnica Salesiana - Campus Sur

terminal. Esto es debido a que se pueden adaptar grippers <sup>10</sup>de diferentes tamaños.

#### **2.1.4 EXACTITUD DE LOS MOVIMIENTOS**

La precisión en un robot industrial obedece algunos elementos que se deben considerar de la siguiente manera:

La resolución del entorno de trabajo, se puntualiza como la extensión del movimiento en la que el robot puede dividir la cantidad de trabajo.

Depende también de dos factores: del control de los sistemas que definen la resolución y los errores mecánicos.

Se define como de control, ya que éste, es el entorno para reconocer todos los aumentos en los movimientos de cada articulación. La controladora que genera la segmentación del intervalo total del movimiento para una junta específica en aumentos independientes (control de mando).

La flexibilidad de dividir el rango de la junta en diferentes lapsos, obedece a la capacidad de almacenamiento en la memoria del dispositivo.

La cantidad de incrementos deducibles para un eje específico es:  $2n$ . Si se tiene, un autómatas con un valor de incremento en  $n = 8$  el controlador puede fraccionar el momento de la transición en 256 estados discretos. De esta manera la resolución del control son intervalos de número (transiciones o movimientos)/256. Los aumentos casi siempre son constantes.

Las fallas mecánicas se encuentran altamente afectadas por la disposición de los materiales que integran las juntas y las articulaciones. Como ejemplos de fallas mecánicas se puede encontrar la amplitud de los engranajes, la rigidez en los motores, los escapes de fluidos, etc.

---

<sup>10</sup> Efector final del Brazo RV-2AJ

La precisión se refiere al volumen de un robot para ubicar el extremo de su articulación en el lugar indicado dentro del entorno de trabajo la cual determina el desplazamiento entre el estado especificado y la colocación real de la articulación final del manipulador.

Sostiene una correspondencia directa con el valor espacial, es decir, con el volumen de control del manipulador de fraccionar el volumen de trabajo.

El manipulador se puede ubicar en el punto de retorno (PR) o en un punto de retorno que posea una distancia equivalente hacia el Primer Punto (PP). Dependiendo del trabajo que se vaya a ejecutar, es la exactitud en la recursividad de los desplazamientos sean estos mayor o menor al proceso inicial, por ejemplo, en trabajos de ensamblaje de segmentos, dicha propiedad será bajo 0.1 ms. En soldadura, pintura y manipulación de fragmentos, la igualdad en la repetición se entiende entre 1 y 3 mm y en los procedimientos de mecanizado, una precisión bajo el 1mm.

La repetición de un punto es usualmente menor que la precisión.

### **2.1.5 VOLUMEN DE CARGA**

La carga, en kilogramos, que puede trasladar mediante el efector final del manipulador, que tiene el nombre de contenido de carga. En ocasiones este dato lo suministran los fabricantes, e inclusive el peso que soporta la pinza.

En estándares de robots industriales, el volumen de la cantidad de carga de la pinza, puede oscilar de entre 1 Kg y 199 Kg en robots de carga media que son los más utilizados. El volumen de carga es una de las particularidades que se consideran prioritariamente en la elección de un robot, acorde a la tarea que se designe. En soldadura y montaje es frecuente estudiar los volúmenes de carga superiores a los 50kg para estos tipos de Robots.

### **2.1.6 RAPIDEZ**

Se refiere a la velocidad máxima dada por las articulaciones. En varios momentos, una aceleración de labor elevada, acrecienta de manera continua el rendimiento del robot, debido a esto la magnitud se evalúa ampliamente en la elección del mecanismo a ser utilizado para el trabajo deseado.

En trabajos de acoplamiento y montaje de piezas es recomendable que la velocidad de labor sea alta. En tareas de matizado, montaje y paletizado, se debe ajustar la velocidad a media o inclusive baja.

### **2.1.7 ACTUADORES**

Son dispositivos substancialmente mecánicos, donde su función es generar fuerza para desplazar o ejercer una acción sobre otro módulo. El impulso que proporciona el actuador deriva de tres orígenes probables: Generación neumática, hidráulica, e impulso electro motriz (motor eléctrico o solenoide).

#### **2.1.7.1 ACTUADORES NEUMÁTICOS**

En estos el origen del arranque es aire a presión entre 4.9 y 9.9 bar. Se generan dos clasificaciones de actuadores neumáticos:

- Cilindros neumáticos
- Motores neumáticos.

#### **CILINDROS NEUMÁTICOS**

Se determina el movimiento de un pistón contenido en un cilindro teniendo como consecuencia, la resultante de presión en ambos sentidos. Los discos neumáticos pueden ser usualmente normal o de doble efecto.

En el normal, el pistón se traslada en un sentido como consecuencia del esfuerzo ejercido por el impulso del aire a presión, mientras, en el otro sentido se traslada en función del efecto de un pistón en el punto en reposo.

En los dispositivos que contienen cilindros de doble efecto, el impulso del aire a presión es el asignado para realizar un empuje al cilindro en las dos trayectorias, al poder ser introducido sin ninguna orientación en cualquiera de las dos cámaras.

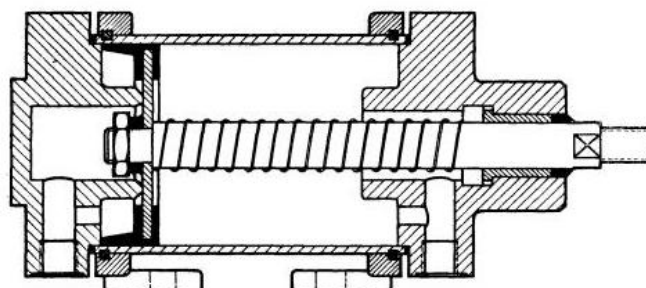


Figura 2.1.7.2.1 Cilindros

Fuente: web3

Generalmente, con los émbolos neumáticos únicamente se necesita una ubicación en los límites del cilindro y no una ubicación continua. Cabe destacar que se puede conseguir una válvula de colocación (comúnmente de efecto continuo) que permite el acompañamiento del aire a presión hacia un lado del embolo alternativamente; se presentan, sin lugar a dudas, estos sistemas de posicionamiento continuo de ejecución neumática, son muy silenciosos y de rápido mantenimiento.

En función de la presión del aire, estos actuadores no logran una buena exactitud de ubicación. Sin obviar, la simplicidad y robustez que hacen apropiado su uso en algunos casos en los que sea necesario, una ubicación en dos contextos disímiles (todo o nada). Por ejemplo, son comúnmente usados en operadores sencillos, en apertura y cierre de sujetadores o en ciertas articulaciones de los robots (se tiene al



desplazamiento derecho de tres grados de libertad de la mayor parte de robots idealmente, SCARA).

Habitualmente debe considerarse que el uso de un robot mediante algún tipo de encendido neumático tenga que disponer de una habilitación de aire comprimido, siendo: un compresor, sistemas de conducción (tuberías, electro válvulas), filtros, secadores, etc. Sin embargo, estas estaciones neumáticas son comunes y constan en algunas de las fábricas donde poseen un mejoramiento en función a la automatización.

### **MOTORES NEUMÁTICOS (DE ALETAS ROTATIVAS O DE PISTONES AXIALES).**

Para estos se obtiene la rotación de un eje a través de aire a presión. De los más destacados se encuentran dos tipos que son, los motores de aspas rotativas y los motores de cilindros céntricos. En los primeros el rotor posee una ubicación que está dispuesta de tal forma que las aspas sean variables en su longitud. Al ingresar la presión en uno de los niveles de la estructura constituidos por dos aspas y la carcasa, tienden a girar hacia un punto en el que el compartimiento posea mayor resistencia.

Los cilindros con bobinados céntricos poseen un eje de giro conjuntamente unido a un tambor que debe girar en función de las fuerzas que realizan algunos cilindros, donde se apoyan sobre un plano transversal.

#### **2.1.7.2 ACTUADORES ELÉCTRICOS**

Las configuraciones de control y simplicidad son los más usados a nivel industrial, siendo esta su principal propiedad ya que está conformado por un sistema autónomo para realizar trabajos lineales de transporte de objetos.

Proporciona una solución efectiva para solucionar antiguos inconvenientes de accionamiento como opción a los cilindros de tipo neumático o hidráulico, ofreciendo un recurso eficiente y más ecológico.

### 2.1.7.3 MOTORES DE CORRIENTE CONTINUA

En función de la practicidad en el control de estos motores, son los más manejados en la industria. Para el control de estos motores, comúnmente se utiliza para el control del motor un sensor de posición (Encoder).

Los motores DC están formados por dos anillados centrales, uno inductor y otro inducido, los cuales se sustentan mediante corriente continua. El inductor, además llamado como enrollado de excitación, está ubicado en el estator y genera un campo magnético de orientación fija.

Conjuntamente, los motores manejados por medio de inducción generan un efecto que mantiene la resolución de giro originado por la generación repetitiva específica que es constante dentro del accionamiento electromotriz. Debido a esto, los tipos de motores DC son controlados por estímulos eléctricos que se usan en el accionamiento con robots.



Figura 2.1.7.4.1 Motores

Fuente: web4

A esta realimentación de la rapidez, se puede agregar otra variable como es el estado de giro, en el cual las revoluciones son tomadas en cuenta con el microprocesador en función del estado deseado y el actual.

El motor de corriente continua se debe llegar a monitorear mediante una constante revisión o cuidado de las escobillas, si el caso lo permite. Cabe destacar que, no es recomendable mantener el rotor parado más de unos minutos, debido al calentamiento que se produce en el colector. Para sortear estos inconvenientes, se suele utilizar motores sin escobillas. Teniendo que, los imanes de generación de energía se ubican en el rotor y el enroscado de inducido en el estator, es posible transformar la corriente a través de mandos estáticos, que obtienen la señal de cambio mediante el detector de ubicación del rotor.

#### **2.1.7.4 MOTORES ASÍNCRONOS DE INDUCCIÓN**

Posiblemente son los más simples y fuertes dentro de la gama de los eléctricos. El rotor se conforma de diferentes barras guías colocadas paralelamente en el eje del motor y por dos abrazaderas conductoras en los extremos. El módulo es equivalente a una celda de ardilla.

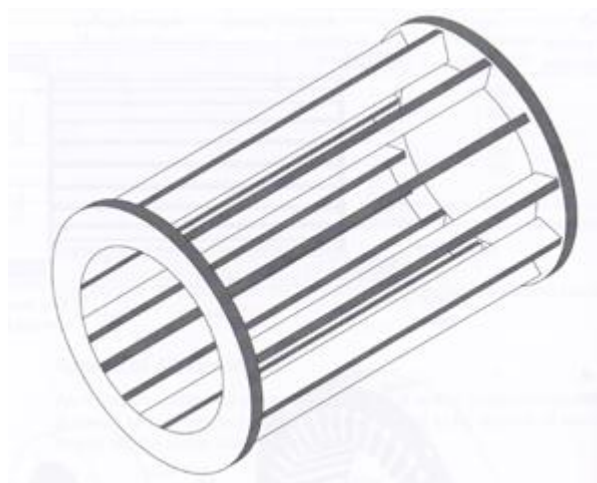


Figura 2.1.7.5.1 Jaula de Motor de Ardilla

Fuente: Web5

El estator contiene un aglomerado de bobinas, debido a esto, mientras la corriente alterna trifásica las traspone, genera una fuerza magnética rotativa en las cercanías del estator de tal manera que conduce la corriente en el rotor, que proporciona su

campo magnético. La disposición entre los dos campos provoca un accionamiento en el rotor. No está establecido un enlace eléctrico continuo entre el estator y rotor.

### **2.1.7.5 MOTORES SÍNCRONOS**

Estos motores, operan equitativamente a una velocidad equivalente al campo del estator, sin deslizamiento.

En los de tipo síncronos la rapidez de revolución del motor depende solamente de la frecuencia en la tensión que mantiene el inducido. Al poder diferenciar la precisión, el control de la rapidez se concreta por medio de un conversor de frecuencias. Para no tener la problemática de pérdida en su sincronismo, se maneja un sensor de colocación estable, que genera la señal de la ubicación del rotor y admite conservar a cada instante el ángulo que constituyen los sesgos del estator y rotor. Este procedimiento de intervención se denomina auto síncrono o automático.

Actualmente se pueden encontrar diversos robots mecánicos que usan este tipo de componentes, con algunas mejoras en los motores DC.

Con relación a los motores asíncronos, no es tan frecuente obtener soluciones de manera satisfactoria a los inconvenientes en el control que llegan a presentarse. Esto ha generado que no se empleen o ejecuten aplicaciones en robótica, en serie.

### **2.1.8 PROGRAMABILIDAD**

La disposición del controlador de tipo microprocesador en los robots para la producción, aloja la programación del robot en diferentes programas.

Los recientes modelos de robots permiten la programación en modo manual, mediante un módulo de codificación.

Las codificaciones combinadas y precisas, manejan varios aspectos del trabajo del manipulador y se pueden componer de la siguiente manera:

- Manejo de la velocidad y la aceleración.

- Datos de programa limitados.
- Control de temporizaciones y pausas.
- Edición, modificación, eliminación y aumento de tareas.
- Mecanismos de seguridad.
- Mecanismos de sincronización con otros dispositivos.
- Programación en lenguajes libres o propietarios mediante un API.

La distribución del manipulador y la correspondencia de algunos módulos generan una disposición mecánica diferente, que origina la definición de las variables que se debe considerar para determinar la ubicación y dirección de la unidad terminal.

Principalmente, constan de cuatro entidades conocidas en los manipuladores, que corresponden con los adecuados tipos de ejes en el espacio y que se mencionan a continuación: cartesianas, cilíndricas, esféricas, angulares.

## **2.2 TIPOS DE BRAZOS ROBÓTICOS**

Existe una amplia clasificación de los brazos robóticos, de los cuales se mencionan a continuación:

### **2.2.1 SCARA (Selective Compliance Assembly Robot Arm)**

El uso de algunas disposiciones en las partes del robot, nota diferentes clasificaciones, con particularidades a tener en cuenta en función del diseño y estructura del robot como en su uso. Las composiciones más notorias son con tres articulaciones, siendo las más importantes al tomar en cuenta el posicionamiento de su extremo en un lugar en el área de trabajo.

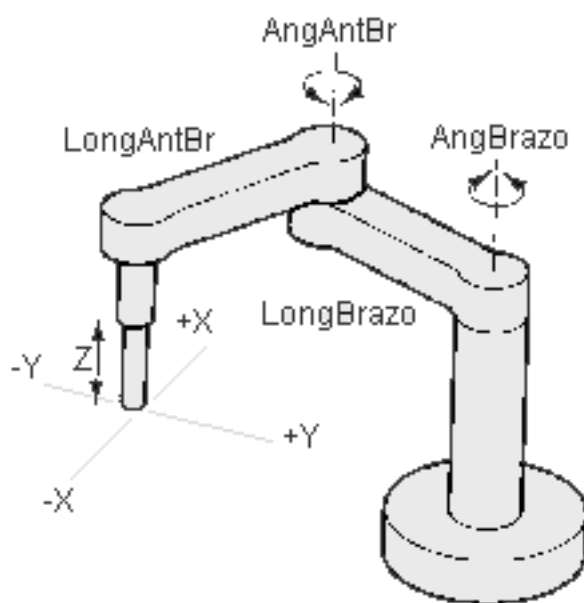


Figura 2.2.1.1 Brazo Robot tipo SCARA<sup>11</sup>

## 2.2.2 CARTESIANOS

La ubicación se hace en el lugar de trabajo con el brazo establecido. Esta configuración se utiliza cuando el lugar de labor es extenso y tiene que ser cubierto, o debido a la precisión que se fundamenta en la perspectiva del robot. Contiene tres desplazamientos lineales, de los tres GDL, que tienen correspondencia con los desplazamientos focalizados en los ejes X, Y y Z.

Los desplazamientos que genera este robot tomando desde un punto a otro, tienen su fundamento en sus composiciones lineales. Interpolación, se define como, el tipo de recorrido que ejecuta el manipulador en función del desplazamiento entre un sitio y otro. Al desplazamiento realizado en un traslado rectilíneo se lo define como inserción lineal y teniendo un recorrido realizado en relación al tipo de trayectoria que poseen sus articulaciones se conoce como interpolación por junta.

<sup>11</sup> Fuente CD FESTO Brazo RV-2AJ

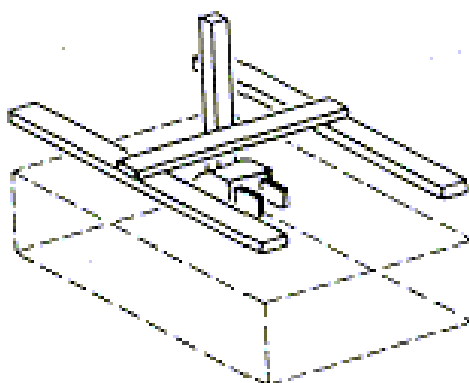


Figura 2.2.2.1 Brazo Robot tipo Cartesiano

Fuente: Web 6

### 2.2.3 CILÍNDRICOS

El robot posee un desplazamiento de giro en su propio eje, una articulación helicoidal para la altura, y una prismática para soporte del radio. El robot se ajusta bien a los lugares de trabajo cónico o cilíndrico. Logra ejecutar dos movimientos, rectilíneo y rotacional, o sea, que exhibe tres GDL

El robot fue construido para elaborar trayectorias definidas como lineales y por articulación. La definida por articulación está en función de la primera junta, debido a que puede realizar un traslado rotacional.

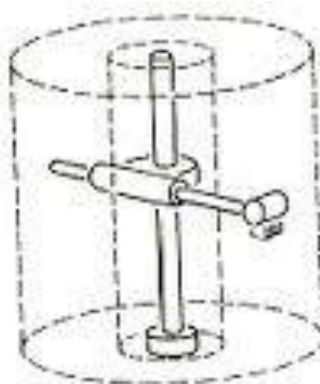


Figura 2.2.3.1 Brazo Robot tipo Cilíndrico

Fuente: Web 7

#### 2.2.4 ESFÉRICO / POLAR

Se tienen dos juntas, de traslación, una para el alcance helicoidal que admite al robot ubicarse en varias trayectorias y ampliar el alcance de la mano a un trayecto circular. Los movimientos comprendidos en este modelo son: rotacional, oblicuo y lineal. El robot maneja la combinación por juntas para desplazarse mediante las articulaciones pequeñas y la principal para el alcance y el encogimiento.

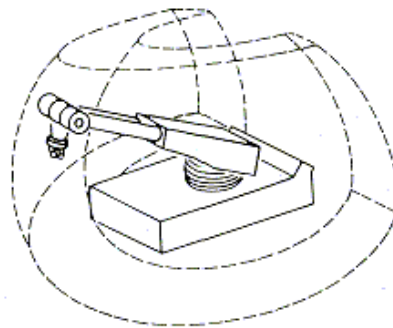


Figura 2.2.4.1 Brazo Robot tipo Esférico o Polar

Fuente: Web 8

#### 2.2.5 ARTICULACIÓN TIPO PUMA

El robot utiliza 3 articulaciones de rotación para ubicarse en el espacio. Usualmente, el cuerpo de trabajo es circular. Este tipo se asemeja al brazo humano. Muestra a la articulación con cambios de posición en rotación y dos angulares. No obstante, el brazo puede efectuar la trayectoria denominada lineal (necesita manejar conjuntamente dos o tres de sus juntas), el desplazamiento real es el generado por la articulación, siendo esta por su rotación y su desplazamiento angular.



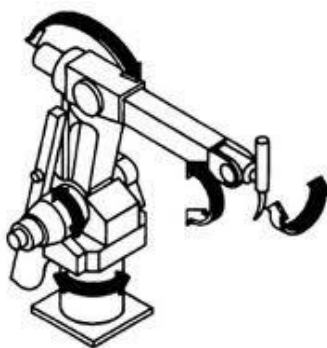


Figura 2.2.5.1 Brazo Robótico tipo Puma

Fuente: Web 9

## 2.3 USO DE LOS ROBOTS INDUSTRIALES

El progreso de los robots industriales desde sus primeros pasos ha sido acelerado. Los test y avances en función de la robótica industrial han contribuido a que los robots abarquen enfoques de desarrollo en algunas áreas productivas en algunos sectores industriales como se revisará más adelante.

En la mayoría de fábricas, los robots logran reemplazar a una persona en aquellas labores monótonas y peligrosas, adecuándose rápidamente a los cambios de fabricación requeridos en función de la demanda que exige el mercado.

La robótica principalmente se enfoca en ampliar los rangos de movimiento, precisión y autonomía de sus operaciones, adicionalmente a ser aptos de conservar una creciente interacción con los usuarios.

Uno de los inventos más útiles en la robótica es su introducción masiva desde el 2002 en las viviendas, teniendo una aspiradora autónoma, conocida como Roomba, y construida por la compañía iRobot.

En Europa, la empresa ABB, se conforma en una de las fábricas más significativas a nivel mundial en la elaboración de robots para la industria, y Suecia como el país con

más robots per cápita. Debido a esto, cerca de 1981 se creó la Federación Internacional de Robótica establecida en Estocolmo.

La fábrica germana KUKA fue ganando terreno en el ensamblaje y establecimiento de robots para la industria, convirtiéndose en una de las principales, en montar una red de soldadura provista principalmente con robots.

Algunos estudios con los robots denotaron ventajas en su uso debido a que estos realizan tareas peligrosas o bruscas para los humanos. Los robots se usan en los trabajos repetitivos y aburridos donde el trabajo de una persona tenga una tendencia a ser ineficiente con el pasar del tiempo.

Los robots logran ejecutar estos procedimientos repetitivos de alta eficacia durante 24 horas al día, sin descansar. Uno de los sectores dentro del proceso de producción que utilizan robots en la manufactura es el automotriz. La compañía General Motors usa alrededor de 10.000 robots para realizar labores como montaje de partes, soldadura, pintura, ubicación y traslado de segmentos metálicos.<sup>12</sup>

El ensamblado es una de las tareas industriales de más vertiginoso desarrollo en la robótica; demanda una mejor exactitud que la ensambladura o la pintura y usa sensores ópticos o catódicos y métodos informáticos de gran estabilidad. Los robots son, utilizados, por ejemplo, en la circuitería de ensamblaje, producción de microchips o en fabricación de módulos electrónicos.

Además son usados en actividades que constituyen un gran riesgo para las personas, como la ubicación de barcos sumergidos en el Océano, exploración de asentamientos minerales submarinos o la excavación de volcanes activos, son fundamentalmente adecuadas para el uso de robots.

Los brazos robóticos generan un acabado de mejor calidad y bajo costo. Además de suministrar plazas de empleo en los campos de programación de software, instalación y mantenimiento de los brazos robóticos, el control de los sensores en la automatización de viejas fábricas y futuro diseño o transformación a una nueva, estas nuevas oportunidades de empleos requieren de mayores niveles de capacidad

---

<sup>12</sup> Ing. Pedro Moreno – Técnico automotriz General Motors -Quito

y formación profesional. En pos del avance y sistematización, las compañías deben afrontar la tarea de facultar a los trabajadores que son desplazados de sus empleos debido a la automatización e instruirles en nuevas habilidades para que logren conseguir un trabajo en las empresas del siglo 21, dando mayor importancia a la parte cognitiva que a la fuerza <sup>13</sup>.

Un modelo de labor conjunta se concentra en los ingenieros trabajadores de la planta de Ringhals, en el suroeste de Suecia, tiene cuatro reactores y produce alrededor del 20% de toda la electricidad utilizada en Suecia. Desde el verano de 2003 descubrieron una importante pérdida de pico en el corazón del reactor. El tubo donde se produjo la falla fue parte del mecanismo de varillas para el control de la seguridad activa durante la clausura del reactor.

Si no se corregía, la falla pudo poner en peligro la seguridad de la operación de la planta sueca, o incluso ser obligado a cerrar para siempre.

La idea era que la tubería iba a durar todo el tiempo en la vida del reactor, y nunca tomaron en cuenta que tendrían que repararlo.

Para acceder al sitio de la pérdida, un ingeniero debe abrirse camino entre una red de cable que se conoce como "la jungla". Si lo hace, podría causar daños irreparables en el reactor.

Este tipo de trabajos confidenciales en lugares de difícil acceso o de mucho riesgo, genera la necesidad de introducir a los robots, dándoles cada vez, más y más confianza.

Los ejércitos los utilizan para desactivar bombas, y existen agencias espaciales que los aprovechan para investigar sistemas siderales.

Los brazos robóticos son mecanismos de operación dúctiles, livianos y se han utilizado en la plaza de estudio científico desde hace 50 años.

Pero a pesar de los esfuerzos de las instituciones más importantes, como la Nasa, siguen siendo principalmente una investigación a nivel de prototipo y a veces solo un producto industrial en áreas muy específicas.

---

<sup>13</sup> Erich Lohrman – Dipl. Ventas Latino América KUKA Alemania

"Han habido varias empresas que me vendieron Robots, pero el mercado para ellos nunca ha sido lo suficientemente grande".<sup>14</sup>

Las compañías ubicadas en Oxford, Reino Unido, han diseñado robots de armas avanzadas, un reactor experimental de manejo nuclear.

En las industrias nucleares francesas y japonesas también son utilizadas para el mantenimiento del interior de los reactores de fusión nuclear.

Uno de los retos más importantes en el desarrollo de estos prototipos son la modelización y el control de sus movimientos.

Un Brazo robótico común, generalmente tiene seis grados de libertad. Esto admite que el alcance puede estar limitado por lo general por una cámara o una herramienta para alcanzar cualquier lugar en el entorno de un brazo normal.

El brazo puede moverse entre algunas posiciones. Este tipo de movimiento es fácil de programar, y es lo que se necesita para la mayoría de aplicaciones industriales.

En lugares estrechos, estos robots solo tienen limitantes con respecto a su tamaño o alcance. Sus articulaciones no pueden realizar los desplazamientos requeridos para conseguir que la punta pueda llegar al lugar deseado sin atascos así que se verá la necesidad de un acople para obviar la necesidad de introducir todo el brazo.

Los nuevos modelos, por otro lado, se pueden mover en 30 direcciones distintas al mismo tiempo. Sus cuerpos ágiles pueden doblarse y flexionarse para hacer frente a la forma del espacio que los confina.

"Hay un gran número de formas en que los brazos se pueden mover. El número de movimientos posibles hace que la matemática sea muy interesante"<sup>15</sup>, se han

---

<sup>14</sup> Joel Burdick, profesor de Robótica en el Instituto de Tecnología de California.

desarrollado programas de ordenador que pueden manejar estas distancias formas serpenteantes de brazos, que permite al operador mover la punta del brazo cuando se necesita.

Una vez revisado algunos modelos de Brazos robóticos, se presenta una clasificación según sus generaciones.

	<b>Robots industriales por descendencias</b>
<b>1ª Generación</b>	Genera la actividad programada por niveles. No se considera las posibles variaciones de su medio de trabajo.
<b>2ª Generación</b>	Obtiene información restringida por su entorno y procede en forma secuencial. Logra ubicar, catalogar (visión) y detectar obstáculos y adecuar sus desplazamientos en su resultado.
<b>3ª Generación</b>	Su clasificación se debe al empleo de lenguaje natural. Tiene la capacidad para la proyección mecánica de sus trabajos.

Tabla 2.3.1 Clasificación de robots industriales en generaciones

---

<sup>15</sup> PHD. Buckingham Academia Real Británica de la ingeniería.

	<b>Jerarquización de los robots acorde a la AFRI</b>
<b>Tipo A</b>	Robot con intervención manual o teach.
<b>Tipo B</b>	Manipulador automático con niveles pre-programados; limitados a través de micro switch o sensores catódicos; controlados por PLC.
<b>Tipo C</b>	Robot programable con recorrido rectilíneo o punto a punto. No posee noción sobre su entorno.
<b>Tipo D</b>	Robot apto para recibir datos de su entorno, reajustando sus tareas en función de las variables.

Tabla 2.3.2 Clasificación de robots industriales según AFRI<sup>16</sup>

### 2.3.1 AUTOMATIZACIÓN EN LA INDUSTRIA DEL AUTOMÓVIL

La industria del automovilismo genera cada vez más oportunidades de producción para los robots debido a que necesitan mayor número de modelos, esto ocasiona directamente la necesidad de generar producción en serie, para esto se adquieren decenas de Robots alrededor del mundo, debido a su precisión y volumen de rendimiento en función de las horas de trabajo.

Ha existido gran movimiento en el mercado mundial durante los 40 años en los que los robots realizan labores para la manufactura automotriz, los estándares caros de los modelos recientes han dado lugar a los robots contemporáneos, más flexibles y fiables de alta tecnología en la actualidad. Con cada nuevo modelo terminado, el proceso de producción va mejorando, paso a paso, en función de la rentabilidad de

<sup>16</sup> (AFRI) Asociación Francesa de Robótica Industrial.

tener un robot. Esto permite justificar su sitio de labor en la actualidad más rápido que antes. La soldadura, fundición, en el corte con láser o paletas, estos equipos mejorados admiten optimizar la cadena de elaboración y de ocupación en las empresas para presionar la producción en función de la aplicación y produciendo un adelanto en el acabado de pintura y acoplamiento final.

En los equipos de derretimiento, se destacan por su robustez y firmeza. Se los identifica por su difícil manejo pero confiable e incansable uso de las pinzas de soldadura pesada. Precisos y potentes para aplicaciones de alta precisión del ensamblaje. Son enfocados a la universalidad de integrar una gama flexible y segura en la mayoría de cualquier asunto de trabajo. Asimismo se tienen los requisitos precisos para poder incorporar aumentos o cambios futuros en la elaboración. Rápido acrecentamiento de la fabricación o la garantía firme en la producción.



Figura: 2.3.1.1 Robots utilizados para ensamblaje de autos

Fuente: Web 10

### **2.3.2 AUTOMATIZACIÓN EN EL ÁMBITO DE LOS PROVEEDORES DE LA INDUSTRIA DEL AUTOMÓVIL.**

Los proveedores son cada vez más importantes para la industria automotriz. Los clientes aportan la mayoría del valor añadido, se toman las ocupaciones de investigación y desarrollo, así como la proyección para la fabricación de componentes y son coautores de gran parte de las creaciones de la actualidad debido a sus sugerencias<sup>17</sup>. En su lugar, los desarrollos de los creadores de automóviles igualmente crecen: la división está establecida por los períodos de avance cada vez más breves, las exigencias cada vez más altas de calidad, productividad, flexibilidad en la producción, la credibilidad y la exactitud en la entrega de suministros. En función a estas circunstancias, no es fácil de mantener cierto grado de libertad creativa, sin embargo, gracias a las soluciones de automatización, es posible. Se atribuye principalmente a la funcionalidad que tiene en todo el mundo que permite a los proveedores comunicar al fabricante, para conseguir satisfacer las demandas del sector automovilístico de una manera óptima y duradera.

Más de la mitad de la producción de cada coche es realizada hoy en las fábricas industriales automatizadas con robots. Esto, no es de raro, debido a que los robots captan cada vez más jerarquía para la fabricación en serie. De los neumáticos a los tubos de gasolina, desde el bloque del motor a las lunas del volante, para el revestimiento de plástico, su trabajo vale la pena en cualquier caso debido a que sus aplicaciones son flexibles, la unidad de control es fácil de manejar y los procesos de producción se ejecutan sin interrupciones, ni necesidad de mantenimiento diario, incluso en condiciones extremas. En combinación con la baja inversión en relación a la producción y la máxima flexibilidad de los costes de mecanizado contribuyen a aumentar la eficiencia del sector y asegurar su futuro en la industria.

---

<sup>17</sup> Ing. Ernesto Andrade Gerente de ventas - Quito Motors



Cuando se trata de posmecanizado estándar de piezas de casi cualquier material, se ha optado por la generación de soluciones robustas y precisas con alto desempeño tecnológico.

Los robots realizan las labores más riesgosas para los empleados, los períodos de trabajo son ejecutados de forma totalmente automática, la relación de los diferentes conjuntos productivos como las estaciones de trabajo y todo ello sin desperdiciar ni un poco de flexibilidad en la producción. Los robots de trabajo precisamente son el factor importante de producción, soldadura, tratamiento de superficie, clasificación, medición y control se llevan a cabo de forma aún más rápida y rentable.



Figura: 2.3.2.1 Robots para producción en serie de ensamblaje de autos

Fuente: Web 11

### **2.3.3 AUTOMATIZACIÓN EN LA INDUSTRIA ALIMENTICIA**

Las grandes compañías de fabricación alimenticia se apoyan en las técnicas de producción de la automatización. La composición de los niveles de automatización en

la secuencia de trabajo, certifica la capacidad en su producción y trabajo de manera permanente. Los robots brindan mejorías determinantes en relación con, la elaboración, apariencia, provisión y logística, para elaborar de un modo confiable, veloz y exacto, inclusive en contextos difíciles de elaboración en la fabricación alimenticia. El proceso de producción es muy cuidadoso ya que el área de trabajo debe estar muy limpia, debido a su aislada forma productiva considerando las rigurosas obligaciones de mantenimiento en la variedad dentro de la protección.

Resisten temperaturas extremas, la rápida transformación y las medidas estrictas de las condiciones de calidad e higiene. Estas son sólo algunas de las muchas demandas que considera la fabricación alimenticia. Bajos costos para el usuario, seguido de una estrecha competencia. Por esta razón, desde el corte hasta el embalaje, a partir de la materia prima, a la industria de la panificación, no sólo deben ser conscientes de la correcta manipulación de alimentos, sino que también debe desplazarse desde un punto de vista empresarial, para lo cual los robots son requeridos en estas etapas de la industria.

También en la industria vale la pena destacar el uso de robots flexibles, ya que descarga las tareas pesadas a la gente y las máquinas. Por ejemplo, en la traslado, montaje o desmontaje de equipos de embalaje en el desposte de carne, en las instrucciones de apilado y paletizado, así como en la acción del control de calidad. Estas son las consideraciones de su uso y se produjeron en función a su labor, confiabilidad, exactitud y con poco deterioro dentro de las horas ininterrumpidas de trabajo, donde sean requeridos. Con ejemplares especiales, como el robot de paletizado que soportan ambientes del Ártico hasta - 30 °C o robots de acero resistente para el corte de las carnes y traslado de tanques de quesos y leche.

Las herramientas en los equipos de fundición, que se destaca por su potencia y resistencia. Se necesita un grado de capacitación para el manejo confiable e incansable de las pinzas de soldadura pesada, son precisas y potentes para aplicaciones de alta precisión del láser.



Figura 2.3.2.1 Robot usado para manejo de alimentos

Fuente: Web 12

#### **2.3.4 AUTOMATIZACIÓN EN LA INDUSTRIA DE LA MADERA**

Para tareas de rectificación de bordes, fresado, taladrado, aserrado, o manipular, cualquier persona que quiera obtener un buen retorno de la apuesta del sector de la madera para los requerimientos de automatización en este ámbito deberá utilizar los Robots. Como consecuencia se logra el máximo rendimiento y un incremento determinante en la productividad. Los robots para el acto de apilamiento y la categorización en serie, a través de la vinculación de grupos provechosos, y como un controlador de calidad, mediante la ayuda de sensores capacitivos u ópticos. En aplicaciones de colocación y recogida (pick & place) son para los multivalentes tareas fáciles de concluir, similar a la perforación de piezas idénticas de alta precisión, como las piezas dobles o lotes. Gracias a su disponibilidad permanente, de práctica programación y su habilidad para ajustarse rápidamente a las nuevas

herramientas y tenazas, los robots pueden contribuir decisivamente a desarrollar la eficacia en la elaboración de corte en la madera.

Los robots pueden garantizar la seguridad de sus empleados y aumentar la producción. Esto también es válido para cualquier tarea perjudicial para la salud, como por ejemplo, la impregnación, el esmalte. Incluso las tareas que implican un alto riesgo de lesiones se llevan a cabo de manera profesional y confiable de los flexibles robots de brazo articulado.

A medida que los robots son capaces de adaptarse rápidamente a los nuevos productos y no se limitan, como las máquinas actuales, para algunos procesos, pueden tener un aglomerado de tareas diferentes. Gracias a su alta precisión, pero aún no se puede asignar tareas complejas de manera autónoma a estos robots como la auto programación a nivel industrial.



Figura 2.3.4.1 Robot con efector adicional para trabajar en madera

Fuente: Web 13

### **2.3.5 AUTOMATIZACIÓN EN LA INDUSTRIA DEL PLÁSTICO**

Los robots en la fabricación de plástico son tomados en cuenta por su flexibilidad de programación y su posibilidad de funcionar en conjunto con los procesos de automatización. Tomando en cuenta la practicidad y mantenimiento eventual, no es difícil decidir sobre estos robots modernos de varias articulaciones.

Estas máquinas adoptan una alta capacidad con una programación funcional y no compleja, y se pueden encontrar tanto en versión para salas esterilizadas para la fabricación de productos químicos y como un modelo de gran exactitud para conseguir la máxima precisión en función de sus labores o para el manejo de materiales peligrosos.

Para permitir a la industria del plástico hacer frente a exigencias extremas a gran escala, esto, no implica ningún problema para los robots, dado que son perfectamente adecuados, tanto para la producción de herramientas con el nivel de calidad requerido por las habitaciones limpias, por el incansable trabajo en la fundición por inherencia. Incluso bajo condiciones muy exigentes de forma fiable aumentar la rentabilidad de los distintos procesos. Debido a que los robots tienen las opciones necesarias para llevar a cabo un sinnúmero de tareas de manipulación, recolección, y la colocación o acabado.

Su capacidad para realizar múltiples tareas que les permite operar de una manera rápida, eficaz y flexible. Gracias a su robustez y larga vida que no se echa atrás ante cualquier necesidad y están muy bien equipados para afrontar a la creciente exigencia de eficiencia y creación.



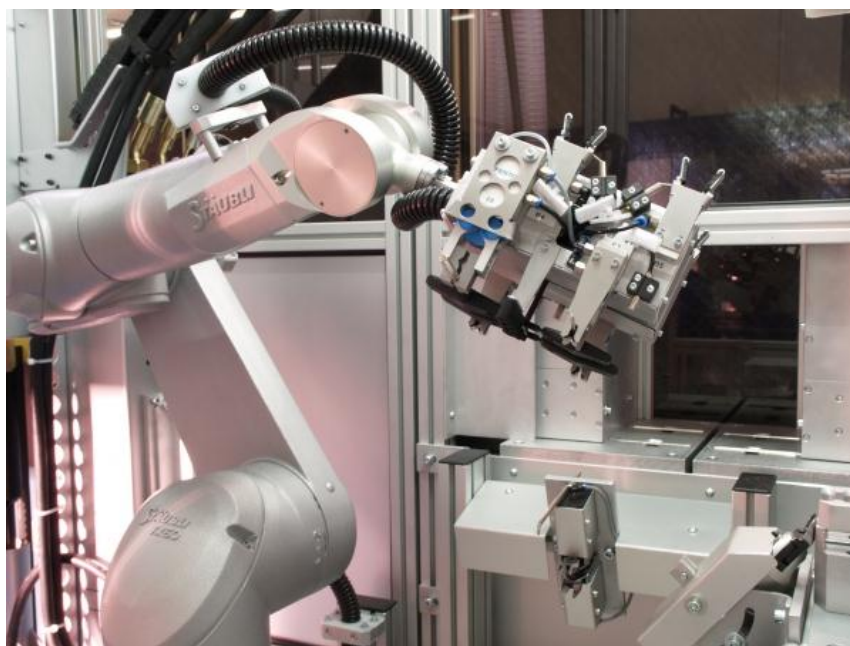


Figura 2.3.6.1 Robot para realizar cortes y torneado plástico

Fuente: Web 14

### **2.3.6 ROBOTS EN LA INDUSTRIA DE LA FUNDICIÓN**

Trabajo en turnos extra en circunstancias extremas, las tareas de hierro fundido son una carga tanto para el trabajador como para la máquina. Una razón más para apostar el trabajo en fuerte función con los robots, ya que están especialmente diseñados para llevar cargas, espacios con un gran nivel de desecho, las altas variaciones de calor y condiciones ambientales adversas. Gracias a la unidad de control fácil de usar y su software modular determinados para la aplicación en el tema, se logra recurrir de manera práctica, ya sea para que controle la máquina de moldeo, entre varios períodos de trabajo o para llevar las piezas muy pesadas. La posibilidad de colocar en cualquier lugar, gran capacidad de carga y fiabilidad cuando se trabaja bajo una carga constante hace que no sólo sean rentables, sino también

los hacen casi imprescindibles en un sector en el que la competitividad debe estar garantizada e intensificada.

Una de las sueldas más utilizadas en el Ecuador es la MIG debido a que esta no produce desperdicio de material y el acabado es más resistente ya que está hecho de un compuesto llamado Hinox, este es un material resistente al frío y al calor y también permite realizar una soldadura de mayor calidad y el período de secado es menor que el resto evitando también desgaste en la suelda en función al tiempo de exposición al medio ambiente.

Incluso cuando se alcanzan altas temperaturas y gracias a su estructura modular, los dispositivos, el control dúctil y los sensores en cuestión, como el de temperatura, sensores catódicos o de ultrasonido, satisfacen los requisitos de la alta competitividad en el entorno de la automatización para el mercado de la fundición.

Los robots también realizan tareas de posmecanizado, tales como el desbarbado, esmerilado y taladrado, o para llevar a cabo controles de calidad.

Teniendo en cuenta que siempre se encarcela a las piezas en el bebedero, se puede manejar distintos productos. El robot posiciona la pieza de hierro fundido en el frontal de los sensores, que se verifica con la ayuda de la unidad de control, si la pieza está completa, continúa su proceso de acabado, si es que realmente existe sino continúa con la siguiente. El robot puede sumergir la pieza en la fundición que se halla entre 250 y 300°C, mientras haya completado un baño de congelación con aprox. -50°C, por otra parte, son evacuados por un plano inclinado a un contenedor.

En el instante en que el Robot Capta la señal de control detecta que la pieza es correcta y en esta es aplicada un rociando en el molde de la máquina con aire a alta presión y un agente de desmoldado soluble en agua, de modo que la siguiente pieza pueda ser fácil de obtener. Dado que el robot mantiene un control mediante un montaje en la máquina con el consiguiente ahorro de espacio, funciona con mucha

precisión, de esta forma la empresa obtiene un beneficio adicional para la reducción del consumo de desmontaje.



Figura 2.3.6.1.1.1 Robot para manejo de materiales en temperaturas altas

Fuente: Web 15

#### **2.3.6.1.1 PIEZAS DE FUNDICIÓN PARA LA PRODUCCIÓN EN SERIE**

En el proceso de fundición se genera la necesidad de tener en cuenta varios parámetros de recolección de señales en las tareas de operaciones, destacados a continuación.

##### **A. ESTADO DE SALIDA / TAREA**

El progreso y presentación de la sucesión de piezas de derretimiento requiere capacidad de maquinaria pesada, que sin duda carece en algunos sectores de la producción. Para superar este problema, se creó, el Centro de Tecnologías de fundición a presión, cuya tarea es la preparación y desarrollo de la capacidad de producción de piezas fundidas de aluminio y magnesio.



## **B. IMPLEMENTACIÓN / SOLUCIÓN**

A través del robot, un átomo estándar de fundición se introdujo en un entorno de producción totalmente automático. Por lo tanto, la capacidad de la producción de la serie se deja ver y si es necesario optimizar, aún más con respecto a todo el proceso, incluyendo la recolección, enfriamiento y corte.

En el momento que se utiliza la máquina de fundición a presión se abre la correspondiente cámara para un robot que elimina el desperdicio del resultado de la fundición. Su ejecución parte desde el inicio de la producción y por esta razón, se puede manejar prácticamente todas las piezas, entre las más grandes. En un siguiente paso, las posiciones de las piezas son colocadas con sensores, a través del cual el controlador comprueba si la pieza está realmente presente y si está completa.

Y transfiere la pieza a una máquina de corte. En la implementación de este proceso, también está integrada la conversión de las señales para la estación, de modo que, si es necesario, la pieza vuelve a ser precargada y reconfigurada en caso de cambio de material antes de pasar la nueva revisión por la prensa.

Otro Robot utiliza para el procedimiento de fundición un Tixotrópico, donde un rayo reemplaza al módulo líquido no newtoniano bañado en aluminio usado regularmente. El robot toma una bandeja que contiene un tornillo, que luego se descarga en la cámara de carga de la máquina, o recoge un remache que es liberado en un pedestal, con la mayor superficie de amarres en toda la longitud del perno y la circunferencia, no se deforma, poniendo más tarde en la cámara, son algunas de las aplicaciones de los Robots en este sector de la producción.



Figura 2.3.7.1 Robot con efector resistente para elaboración de metales en serie

Fuente: Web 16

### **2.3.7 ROBOTS QUE CIERRAN HERMÉTICAMENTE**

Estos productos gozan de gran popularidad y con el fin de ser igual a la demanda, las compañías cuentan actualmente con plantas embotelladoras ultra-modernas y altamente automatizadas.

Una de ellas es muy especial, es decir, de acero inoxidable, un robot determinado para otros procesos donde el cliente requiere un robot para cerrar y para paletizar barriles.

Existen otra gama de Robots que brindan una solución compuesta para el Robot de soldadura. Esta estructura brinda algunas mejoras sobre una figuración llamada robot convencional.

La formación de emanación de la antorcha elimina el cordón típico del acabado, se puede acrecentar la duración y la fiabilidad del módulo del cordón de la antorcha.

El aumento de la densidad del arco es fácil de lograr con dos o más robots trabajando en estrecha colaboración sin interferencias.

Una posición corta de la antorcha en combinación con una línea coaxial de la pinza maximiza el trabajo de rotación utilizable en el robot. La antorcha del brazo se elimina, y la soldadura circular se optimiza.

Los robots industriales y las soluciones de automatización deben cubrir una amplia gama de aplicaciones para la industria aeroespacial, automotriz, bienes de consumo, de alimentos, fabricación de metales, médica, farmacéutica, energías limpias y muchas otras industrias.

La Empresa FANUC Robotics en América es una de las principales ensambladoras de robótica para la industria con un alto porcentaje de robots instalados en el continente y en mayor porcentaje alrededor del mundo. Estos robots tienen una capacidad de carga que van desde Paquetes de 0,8 kg a 1250 kg.

En conjunto con el software de la aplicación, la precisión en la detección y sensores catódicos, generan una reducción de costos y aumentan su ventaja competitiva.

Se ha desarrollado y puesto en marcha un sistema compuesto por un Robot Kawasaki FS20N, una herramienta de corte de alta velocidad y una mesa giratoria.

Alguna de las operaciones necesitan una herramienta de corte, para mejorar su producción por lo que se simplifica el trabajo y acelera la entrega del producto.



Figura 2.3.7.1 Robots industriales Fanuc

Fuente: Web 17

### 2.3.8 ROBOTS PARA CORTE

El corte de piezas con la ayuda de robots, es un procedimiento reciente que merece mucho interés. La modularidad en la reprogramación del robot y su unificación en un sistema CIM<sup>18</sup>, permite que sea el complemento para trasladar el instrumento de corte sobre la pieza, realizando con precisión su tarea de corte.

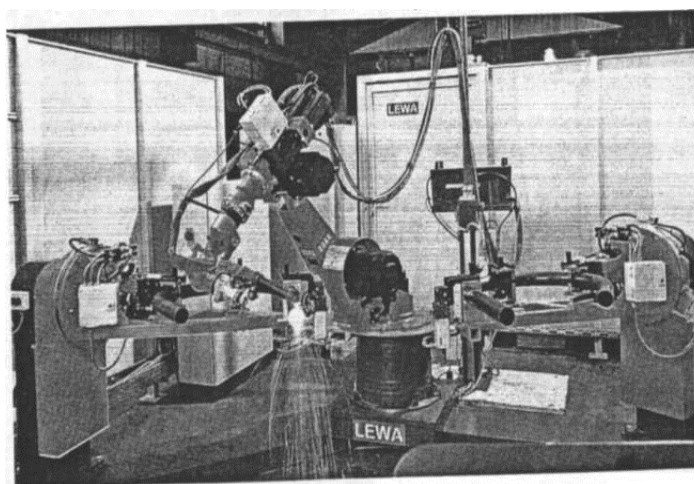


Figura 2.3.8.1 Estación de corte por plasma robotizada. (Kuka).<sup>19</sup>

<sup>18</sup> Manufactura integrada por computadora

<sup>19</sup> Catalogof ideas - KUKA

### 2.3.9 ROBOTS EN LA CONSTRUCCIÓN

En la división de la construcción se utilizan para apilar o realizar hoyos en la tierra o desplazar cerramiento, como por ejemplo en España y Alemania, como los países en los que se movilizan mayor cantidad de recursos económicos y tecnológicos

Se destaca la existencia de un notable interés en el progreso de sistemas automáticos, situados en automatizar en lo posible algunas de las variadas labores que forman parte de la construcción.

Entre los probables trabajos que podría incluir robots se pueden destacar la construcción de edificios (centros comerciales, o conjuntos habitacionales de tipo prefabricado).



Figura 2.3.9.1 Asistente de soldadura de cerramiento

Fuente: Web 18

### 2.3.10 Robots en la Industria nuclear

Debido al tratamiento especial, la división nuclear es una de las más aptas para el uso de robots definiendo diferentes diseños. Entre las diversas aplicaciones se tienen procedimientos de mantenimiento en zonas de desechos y de manejo de restos nucleares.

Los controles de operaciones de reconocimiento y mantenimiento de perímetros contaminados de una central nuclear de elaboración de energía eléctrica, son comúnmente largas y costosas de realizarlas manualmente, el período de exhibición de los técnicos a la radiación es un componente crítico que, junto con el eminente costo que agrega una detención temporal del funcionamiento del sistema, demuestra, la necesidad del uso de sistemas robotizados.



Figura 2.3.10.1 robot para asistencia nuclear

Fuente: Web 19



### **Robots aéreos y submarinos**

Estos son comúnmente transportes independientes, aptos para movilizarse por sus pertinentes medios, transportando una carga destinada al progreso de un cometido concreto.



Figura 2.3.10.2 asistencia militar

Fuente: Web 20

### **Robots asistenciales**

El requerimiento de cuidar y apoyar al creciente número de personas mayores de edad y el interés de apoyar con un mayor grado, para dar la independencia posible a

las personas que por alguna razón tienen la movilidad limitada, admite que el progreso de robots asistenciales obtenga un alto interés.

Las sillas de ruedas parciales o totalmente autónomas, agregan a las sillas motorizadas, control de navegación y vigilancia, incorporando sensores para un mejor manejo del entorno así como también control de cercanía de un obstáculo o hueco o sistemas de predicción de cruce de peatones o vehículos.

### **Robots para el ocio**

En esta categoría se encuentran los robots de servicio, se abarcan todos los usos en los que el robot aporta en actividades de diversión y ocio de las personas, algunas actividades se pueden destacar:

Robots ayudante para restaurantes.

Robots para museos (Animatrónica).

Robots para parques de diversiones (Robocoaster).



Figura 2.3.10.3 Robocoaster<sup>20</sup>

---

<sup>20</sup> Catalog of ideas KUKA



## 2.4 UTILIZACIÓN DE BRAZOS ROBÓTICOS EN EL ECUADOR

Los brazos Robóticos a nivel del mundo han alcanzado en estos últimos años grandes escalas de utilización en función de los requerimientos del mercado debido a los altos riesgos de seguridad, diversos protocolos y estándares de calidad, que a la vez llevan a generar procesos de producción más acelerados.

Es por eso que la tecnología robótica se convirtió en un cambio esencial en la industria, se han generado controversias a razón de la manipulación de la información con respecto a los despidos intempestivos por el reemplazo estas herramientas tecnológicas de alta capacidad y sin horarios de trabajo, siendo estos uno de los principales problemas a nivel global, pero se han realizado estudios, al respecto en la comunidad Alemana<sup>21</sup>, de que no se perderán empleos en función de las adquisiciones de estos multiarticulados sino que las personas regresarán a trabajos donde exija mayor capacidad intelectual y desenvolvimiento con la gente y no perder el tiempo en trabajos repetitivos o en la carga de pesos que podrían invalidar a una persona, es por esto que los brazos robóticos han tenido gran acogida en el mercado.

En la última década también se han proyectado una gran cantidad de brazos robóticos a nivel didáctico, se debe a la gran relación que existen con los brazos industriales a nivel de capacitación de la gente, la familiarización con estos equipos educativos permiten tener un lenguaje más familiar a la hora de conseguir un brazo industrial debido a que la manera de programar es la misma, claro dependiendo de los lenguajes que manejan las casas comerciales, pero ya es un primer paso para lograr entender los principios fundamentales de introducir estos brazos al mercado.

En el mercado Ecuatoriano realmente no se ha logrado introducir una política de robotizar debido a diferentes factores, como por ejemplo, el desconocimiento de estas herramientas en bien de la producción, la falta de estándares en nivel de calidad, el régimen de politización de intercambiar la mano de obra "barata" de una

---

<sup>21</sup> Erich Lohrman, Dipl. Alemán – Representante de KUKA latino América

persona por la de un brazo robótico, los altos costos que alcanzan estos equipos, pero no se analiza las altas ganancias que se generan en el mercado, al acelerar los procesos, donde se han realizado estudios de amortización de estos equipos en aproximadamente 2 años<sup>22</sup>, y la aceleración en la producción, dando inclusive posibilidades de exportación de los productos a nivel masivo, los cuales producirían mayores ganancias, además se han visto casos en los cuales los gerentes o dueños de empresas prefieren adquirir vehículos inclusive más costosos que los mismos brazos, y no se dan cuenta que con ese valor podrían tener una planta de producción más eficiente y que a la larga produciría más réditos que el trabajo simple y convencional que se tiene en la actualidad.

## **2.5 ROBÓTICA EN EL ÁMBITO EDUCATIVO**

Se registran algunas universidades en la cuales han adquirido brazos robóticos para la utilización educativa, entre las cuales se encuentran:

La Universidad Tecnológica Equinoccial, la cual utiliza el Brazo Robótico RV-2AJ para fines académico y prácticos, los estudiantes expresan que existe un gran interés en relación a la adquisición de este multiarticulado para la facultad de mecatrónica en el año 2010.

La Escuela Politécnica del Ejército, hizo la adquisición de 3 Brazos Robóticos Industriales KUKA KR5, Los cuales están a disposición de los estudiantes para tareas de programación y elaboración de proyectos.

La Universidad Politécnica Salesiana Campus-Sur se abasteció de un laboratorio en el cual tiene a disposición dos Brazos Robóticos RV-2AJ, entre otros módulos.

Los cuales se usan para el desarrollo de aplicaciones, para el manejo de brazos MITSUBISHI RV-2AJ.

Las funciones necesarias para la interacción y el desplegado de objetos, pero se basa en una librería que previamente se creó; esta incluye el dibujo de cada uno de

---

<sup>22</sup> Erich Lohrman, Dipl. Alemán – Representante de KUKA latino América

los eslabones del robot, las variables manipuladas en la ejecución del programa y las ecuaciones que se calculan en la fase de la cinemática.

La función que se puede nombrar como fundamental, es la encargada de realizar las operaciones que efectúan cálculos específicos donde los datos son usados para mostrarlos en pantalla y de manera más importante para posicionar a cada eslabón.

Dentro de los brazos Robóticos con mayor expectativa de venta se encuentra un sistema de robot industrial con dos brazos robóticos, suministrado con módulos de visión, optimizado para clasificación y manufacturación.



Figura 2.5.1 Robots industriales Fanuc

Fuente: Web 21

Este sistema, cuenta con un par de brazos robóticos de varios ejes, que le permiten tomar hasta 120 objetos por minuto, a medida que se mueven por la banda transportadora. Sus brazos mecánicos pueden trabajar sin parar 24 horas al día, y son invulnerables al ácido y a la corrosión de composiciones alcalinas.

Las industrias nacionales enfocadas en la robótica son pequeñas y están en la categoría de microempresas.

Estas empresas tienen que considerar que el hecho de que los autómatas que funcionan en los campos de automatización industrial del Ecuador son importados, especialmente desde Japón, Corea, Estados Unidos, Alemania, México, Brasil. El Ing. Nelson Sotomayor, catedrático de Automatización y Control Industrial de la Escuela Politécnica Nacional, mantiene que es muy difícil que los robots acoplados en el país pertenezcan al mercado nacional.

"Para masificar la venta de robots hace falta apoyo económico por parte del Estado e interés para invertir de parte de la empresa privada", con palabras del Ing. Sotomayor, quien resaltó que estas, " serían inversiones a muy largo plazo, por lo que muchas empresas prefieren no arriesgar sus recursos en productos que resultan prácticamente, el citó<sup>23</sup>".

Sotomayor, concluyó también que "existe la idea de que, si se automatizan los procesos industriales en fábricas y empresas, se eliminarán fuentes de empleo. Sin embargo, al implementar la robótica en estos procesos, la eficiencia de una empresa tiende a aumentar en al menos un 100%, lo que implicaría que las ganancias se dupliquen y que se puedan implementar nuevas plazas de trabajo en pequeñas y medianas empresas que trabajen en conjunto con las grandes".

El gasto para el diseño y montaje de un robot promedia los \$60.000 usd; hasta obtener un modelo final dispuesto para ser copiado y distribuido. El valor de la investigación para obtener el diseño final es del 75% del valor total, debido a que, en la generalidad de procesos, hay que sustituir materiales en las pruebas.

---

<sup>23</sup> Fecha de la entrevista 15/11/2011 Ing. Sotomayor profesor de la facultad de mecánica intrustrial de la EPN

Bajo estas circunstancias, los precios comerciales de los robots producidos en el país promediarían los \$75 000, "cantidad que resulta inaccesible para muchos, mas no para grandes empresas que están en proceso de automatización.

Tecnologías STARBOT, una de las empresas del mercado de la robótica en el país, cuyos propietarios afirman que aún no venden directamente robots, sino "proyectos con robots", pero su proyección es diferente para un futuro, ciertamente definen que la necesidad de robots para la industria es su principal proyección de ventas.

En otro aspecto, Carlos Montero, propietario de Druid, una microempresa que provee robots para automatización industrial, afirma que, en 2008, se vendió un solo robot a una fábrica de montaje de camiones en Ambato, a un valor de \$45000 usd. Agrega que este año no han recibido pedidos, siendo esta la realidad de muchas empresas especializadas en el país, por lo que se están planificando nuevos proyectos.

### 3 ANÁLISIS Y DISEÑO

Se presenta la necesidad de generar parámetros reales del sistema que se va a controlar, para ello es imperativo tener una descripción del software y hardware que se va a usar.

#### 3.1 DESCRIPCIÓN DEL HARDWARE Y SOFTWARE

Se establecen los parámetros y dispositivos que se toman en cuenta para la realización tanto del sistema como con la interacción del Brazo RV-2AJ

##### 3.1.1 HARDWARE DE CONTROL

Se realizará una descripción de las capacidades de desplazamiento Robot.

##### 3.1.1.1 MODOS DE DESPLAZAMIENTO

El Brazo RV-2AJ tiene diferentes tipos de desplazamiento, de los cuales se destacan algunos, por ser los de mayor utilidad al momento desplazarse.

- **MODO JOINT**

Consiste en desplazamientos angulares por medio de sus articulaciones en movimientos graduales y mediante el establecimiento correcto de la velocidad, cada junta (J) se mueve independientemente.

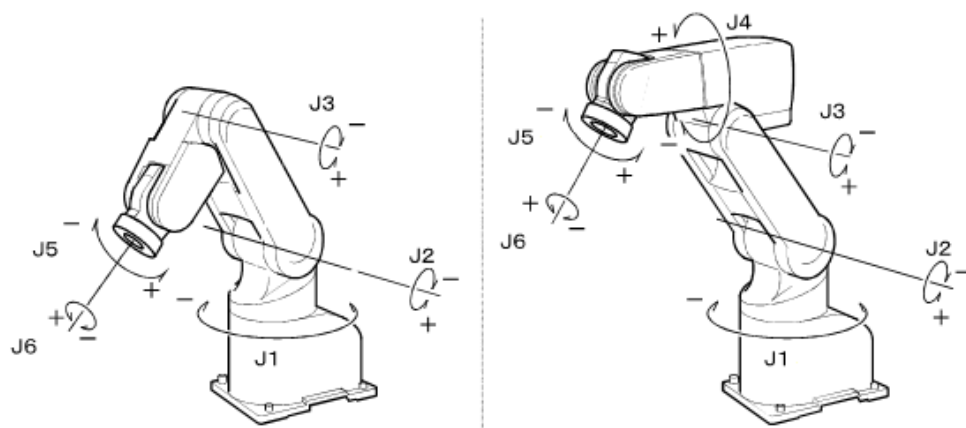
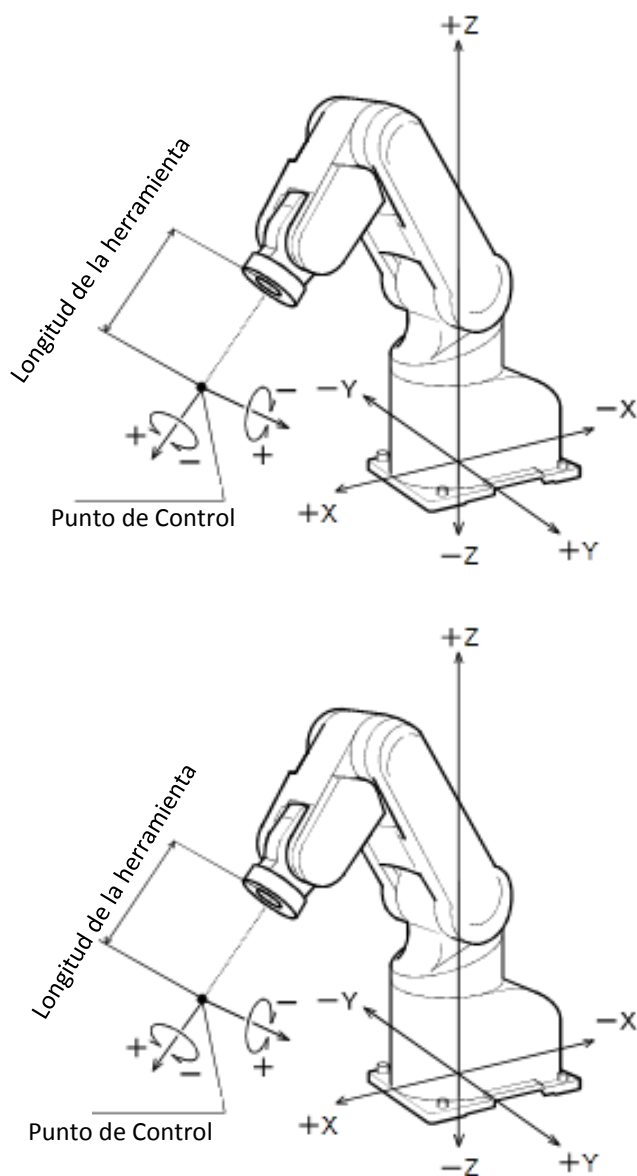


Figura 3.1.1.1.1 Ángulos de alcance en modo JOINT <sup>24</sup>

<sup>24</sup> Fuente CD FESTO Brazo RV-2AJ

- **MODO XYZ**

Permite tener desplazamientos en función del eje de coordenadas XYZ para tener mayor alcance entre los desplazamientos que se desee obtener para utilizar el punto de control



Figuras 3.1.1.1.2 y 3.1.1.1.3 Ángulos de alcance en modo JOINT Controladora<sup>25</sup>

<sup>25</sup> Fuente CD FESTO Brazo RV-2AJ

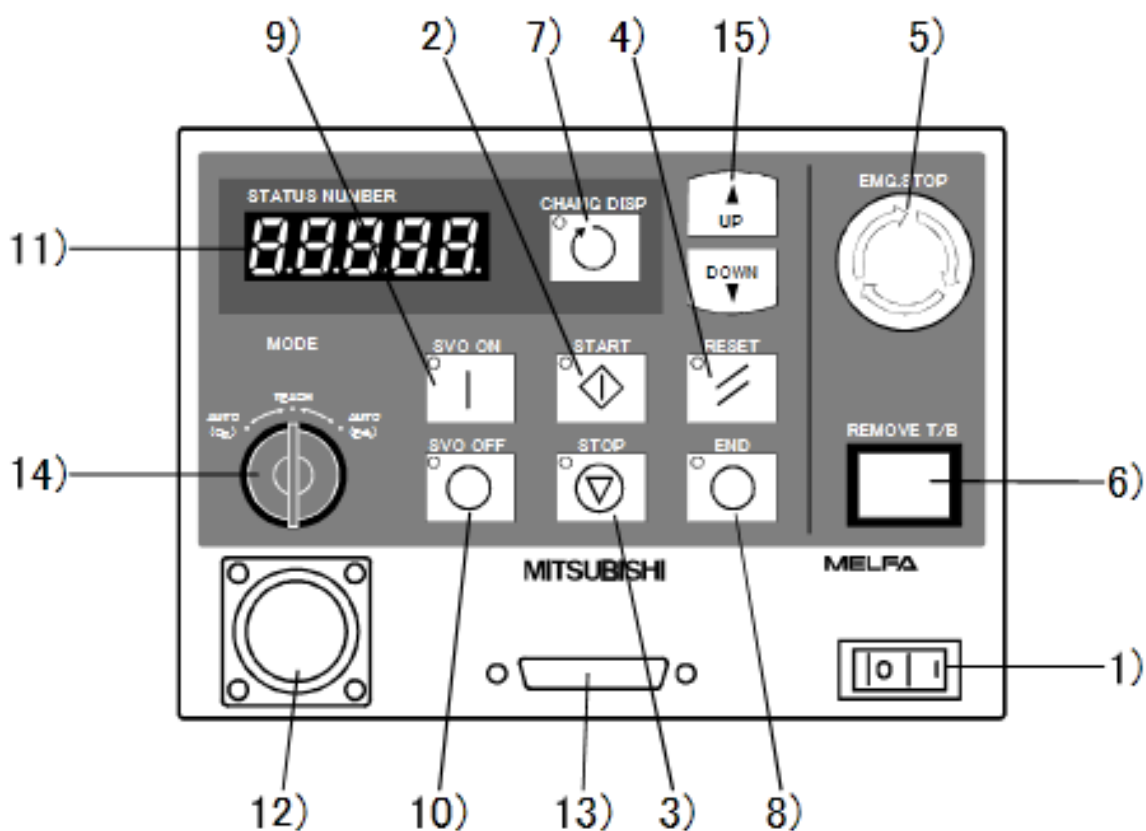


Figura 3.1.1.1.4 Controladora <sup>26</sup>

### Controles para el modo XYZ

1) Interruptor de encendido

Esto convierte la energía de control en ON / OFF.

2) Botón START

Ejecuta el programa y opera el robot. El programa se ejecuta de forma continua.

<sup>26</sup> Fuente CD FESTO Brazo RV-2AJ



### 3) STOP

Evita que el robot continúe desplazándose de inmediato. El servo no se apaga.

### 4) Botón RESET

Restablece el error.

### 5) Interruptor de parada de emergencia

Evita que el robot este dentro de un estado de emergencia. El servo se apaga.

### 6) T / B <sup>27</sup> Switch eliminar

Esto se utiliza para conectar / desconectar la T / B sin desconectar la alimentación del controlador de control.

### 7) Botón CHNGDISP

Cambia los detalles que aparecen en la pantalla en el orden de "anulación" " Programa No. "Línea Número".

### 8) Botón END

Detiene el programa que se ejecuta en la última línea o instrucción END.

### 9) SVO.ON botón

La alimentación del servo cambia de estado a ON. (El servo se activa.)

### 10) SVO.OFF botón

Desactiva la potencia del servo. (El servo se apaga.)

### 11) NÚMERO DE ESTADO (panel de la pantalla)

Despliega No. de La alarma, N ° de programas, valor de error (%), etc., se muestra.

### 12) T / B de conexión

Conector dedicado para la conexión de la T / B.

### 13) Conector del ordenador personal

Este es un conector de especificación RS-232C para la conexión de la computadora personal.

---

<sup>27</sup> T/B (TEACH)

#### 14) Interruptor MODO

Esto cambia el modo de operación del robot.

##### AUTO (Op.)

Sólo las operaciones del controlador son válidas. Operaciones para las que el modo de operación debe estar en el dispositivo externo o T / B, no son posibles.

##### TEACH

Cuando el T / B es válida, sólo las operaciones de la T / B son válidas. Operaciones para las que el modo de operación debe estar en el dispositivo externo o responsable del tratamiento no son posibles.

##### AUTO (Ext.)

Sólo las operaciones de los dispositivos externos son válidas. Operaciones para las que el modo de operación debe estar en el T / B o el controlador no es posible.

#### 15) ARRIBA / ABAJO

Se desplaza hacia arriba o hacia abajo los detalles que aparecen en el "ESTADO. NÚMERO" en la pantalla del panel.

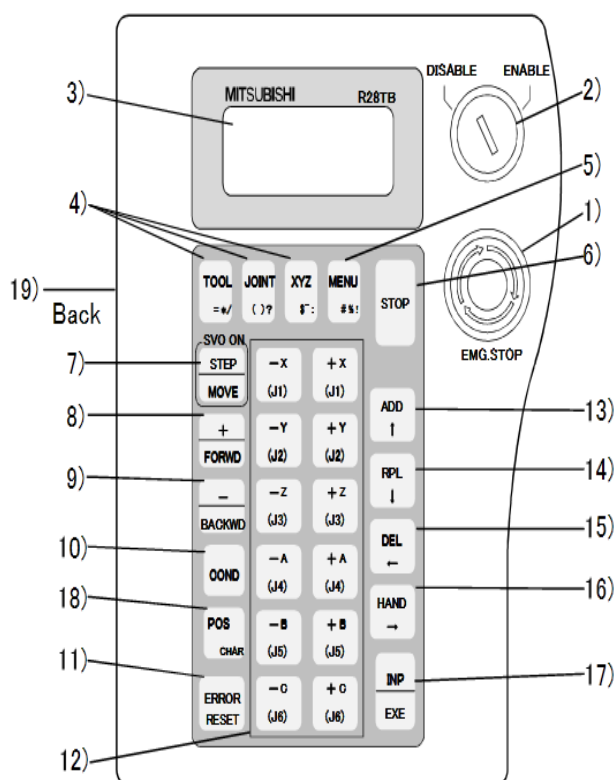


Figura 3.1.1.1.5 TEACH <sup>28</sup>

### 3.1.2 TEACH

#### 1) Interruptor de parada de emergencia

El servo del robot se apaga y la operación se detiene inmediatamente.

#### 2) T / B activar / desactivar el interruptor de cambio

Esta opción cambia la operación T / B permiten la llave de operación entre habilitar y deshabilitar la T/B.

#### 3) Pantalla LCD

Muestra el estado del robot y de los distintos menús.

#### 4) <TOOL, JOINT, XYZ> key

En esta forma se selecciona el modo manual (JOINT, XYZ, de 3 ejes XYZ, cilindro, y Tool).

<sup>28</sup> Fuente CD FESTO Brazo RV-2AJ

5) <MENU> key

Esta función permite una visualización de la pantalla de las diferentes opciones que posee el brazo, tanto modos de funcionamiento como programas.

6) <STOP> key

Esto evita que el programa haga colisionar al robot y desacelera el robot a una parada.

7) <STEP/MOVE> key

Se lleva a cabo cuando se pulsa esta tecla simultáneamente con la del mando. Esto también convierte al servo ON y lleva a cabo saltar un paso.

8) <FORWD> key

Lleva a cabo el paso de alimentación

9) <BACKWD> key

Este cambio lleva a cabo un paso (ida y vuelta por el camino de la operación)

10) <COND> key

Esto reestablece el programa.

11) <Error Reset> key

Esto restablece el error, y libera el límite de software.

12) < Operación Jog> key

Opera el robot de acuerdo con el modo manual. Al introducir valores numéricos.

13) <ADD/> key

Registra los datos de posición. También mueve el cursor hacia arriba.

14) <RPL/> key

Corrige los datos de posición. También mueve el cursor hacia abajo.

15) <DEL/> key

Elimina los datos de posición. También se mueve el cursor hacia la izquierda.

#### 16) <HAND/> key

Abre y cierra la mano. También se mueve el cursor hacia la derecha.

#### 17) <INP/EXE> key

Introduce el programa, y lleva a cabo el paso alimentación / retorno.

#### 18) <POS Char> key

Esto cambia la pantalla de edición, y los cambios entre los números y caracteres alfabéticos.

#### 19) <DeadMan switch>

Cuando el Teach tiene el estado de [Activar / Desactivar]

### 3.1.3 MANIPULADOR

A nivel mecánico, el efector final es el principal elemento de un brazo robótico, el cual se conforma por una serie de componentes estructurales sólidos unidos mediante articulaciones que permiten un movimiento relativo entre cada dos eslabones consecutivos.

Las partes que pertenecen al manipulador se denominan, entre otros con los nombres de: cuerpo, brazo, muñeca y actuador final (o elemento terminal); este último se le conoce como ariete, sensor, garra, pinza o gripper.

### 3.1.4 EFECTOR FINAL O GRIPPER

El efector final es la herramienta que se encuentra en el extremo del brazo robótico y le permite interactuar con el entorno. El modelo RV-2AJ tiene la capacidad de utilizar Grippers neumáticos y eléctricos. Dependiendo del grado de apertura del Gripper varía su capacidad de carga, según la siguiente gráfica:

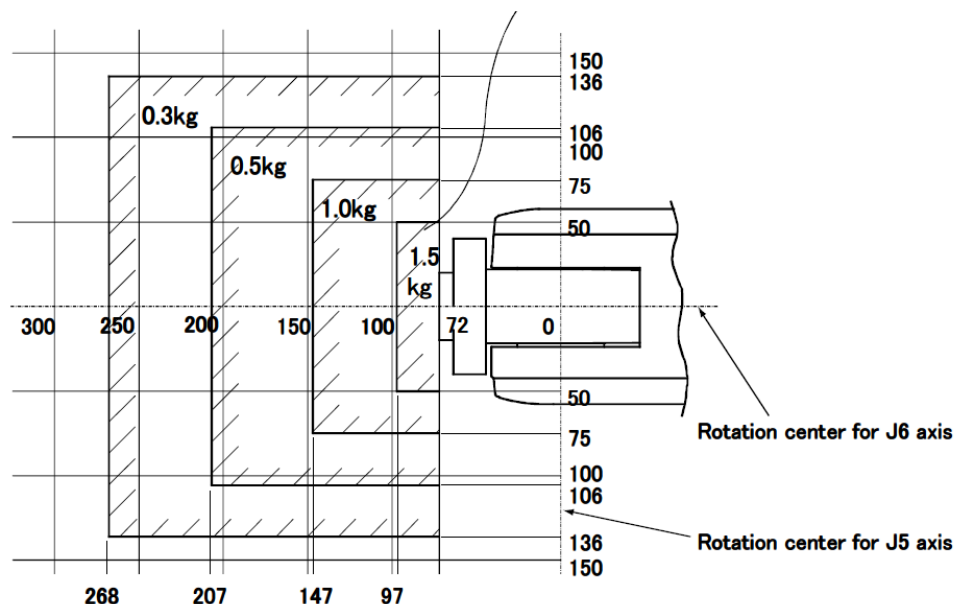


Figura 3.1.4.1 Capacidad de carga del Gripper<sup>29</sup>

La orientación de un eslabón del manipulador se determina mediante los elementos roll, pitch y yaw

A la muñeca de un manipulador le corresponden los siguientes movimientos o grados de libertad: giro (hand rotate), elevación (wrist flex) y desviación (wrist rotate).

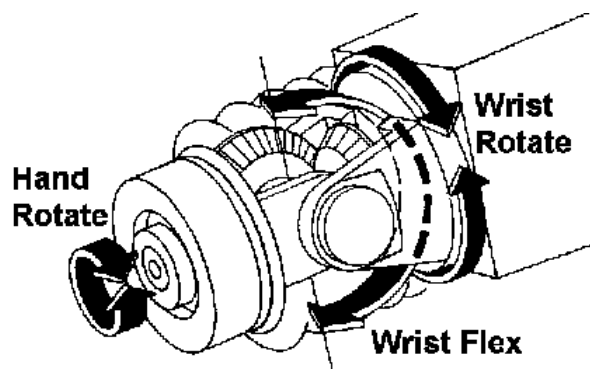


Figura 3.1.4.2 Muñeca de acople para la pinza<sup>30</sup>

<sup>29</sup> Manual Festo 175260

<sup>30</sup> Manual Festo 175260

### 3.1.5 RANGO DE OPERACIÓN

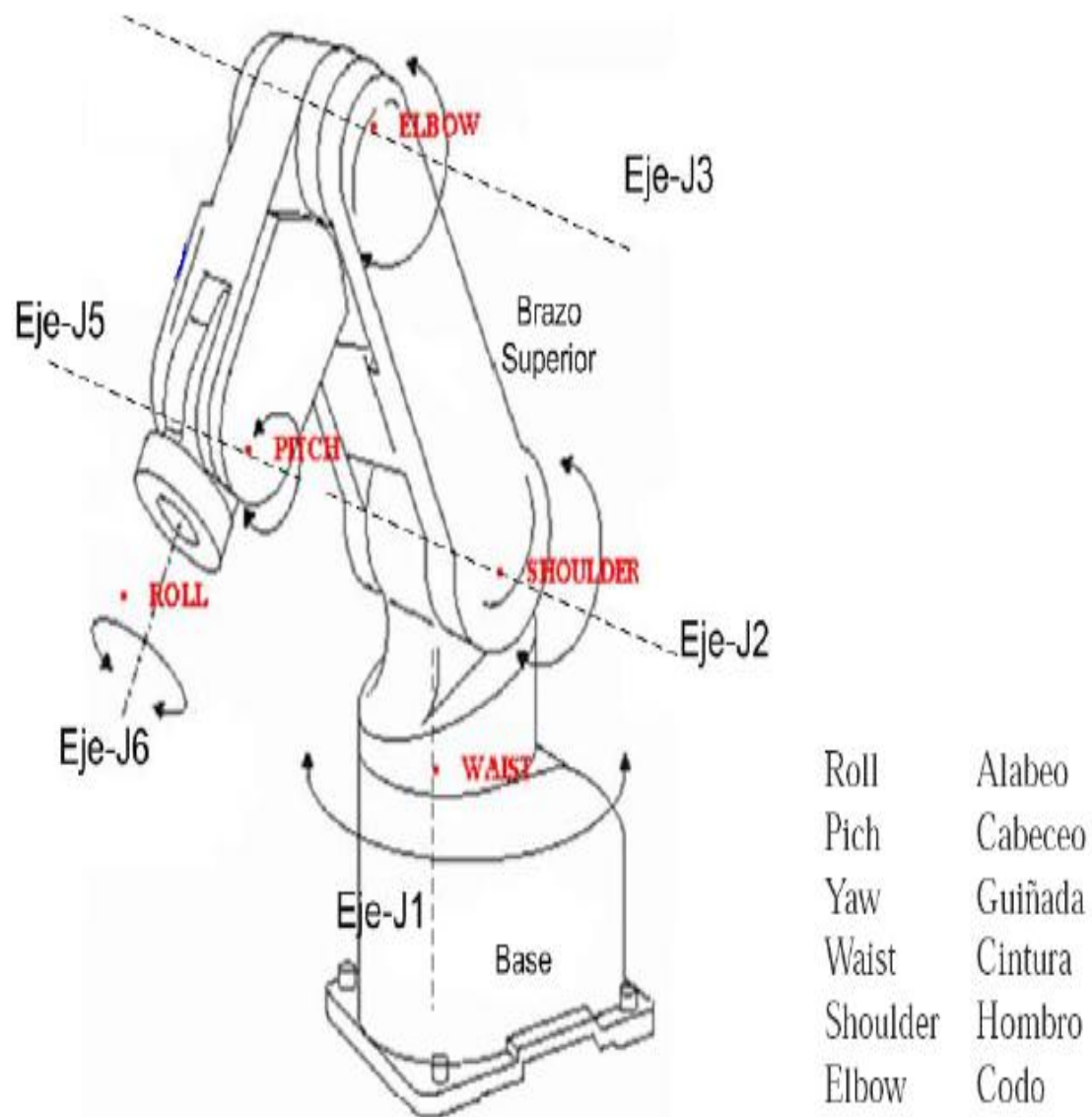


Figura 3.1.5.1 MODO JOINT<sup>31</sup>

<sup>31</sup> Manual CD FESTO

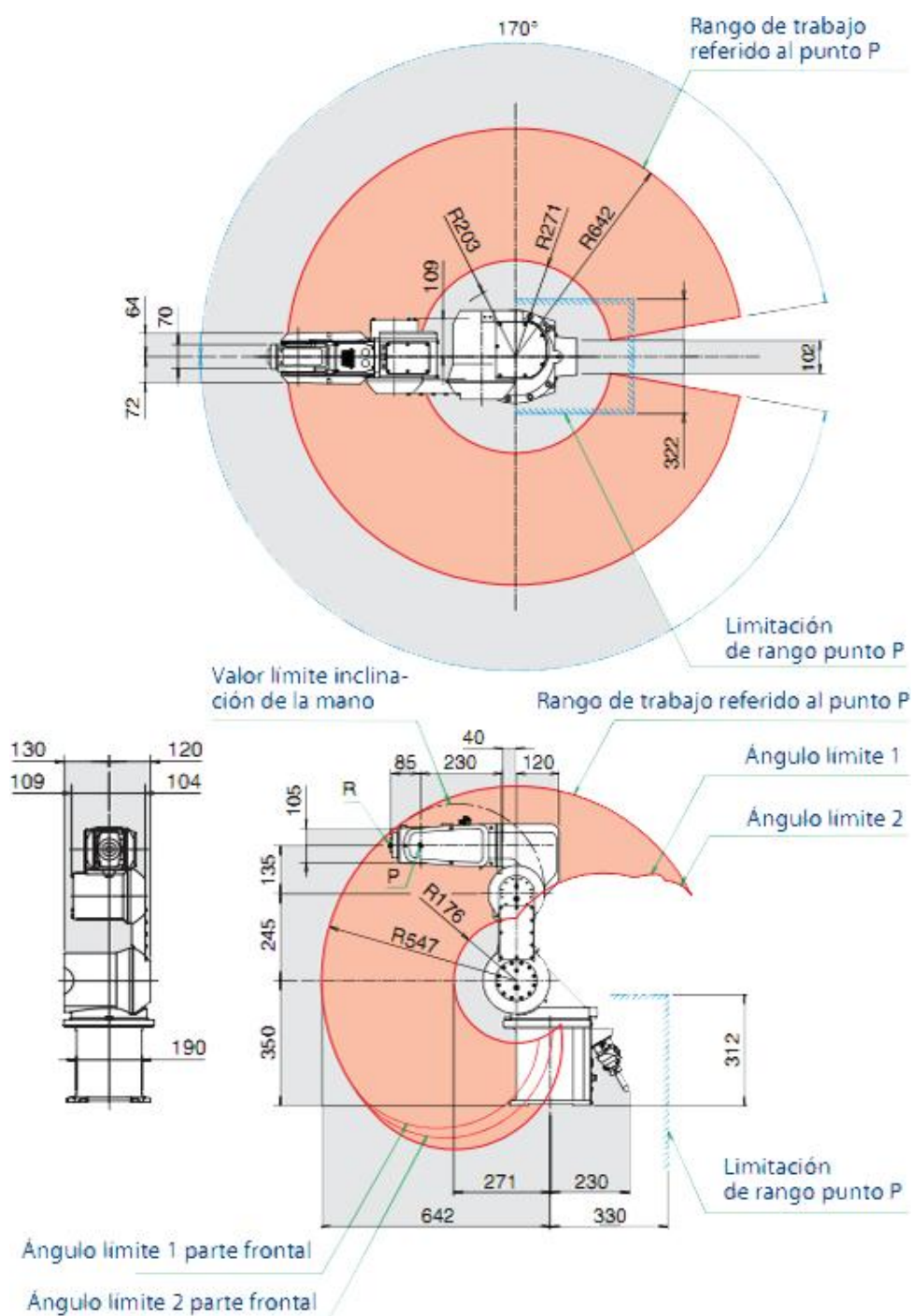


Figura 3.1.5.2 Esquema de movimiento en MODO JOINT<sup>32</sup>

<sup>32</sup> Manual CD FESTO



### 3.1.6 ESPECIFICACIONES TÉCNICAS

Modelo		RV-1A	RV-2AJ
Movimiento de los grados de libertad		6	5
Driver de Motor		AC servomotor (J1-J3 y J5 angulo de freno asignado)	
Detección de detección		Encoder Absoluto	
Carga Máxima (kg)		1,5(1)	2(1,5)
Longitud del Brazo (mm)		250+160	250+160
Rango máximo de Radio (mm)		418	410
Límites de Operación	J1	Deg	150 Aprox
	J2		180(-60+120)
	J3		95(+60+155)
	J4		230(-110+120)
	J5		160 Aprox
	J6		-----
Velocidad Máxima	J1	Deg/s	90 Aprox
	J2		200 Aprox
	J3		180
	J4		90
	J5		135
	J6		180
Máxima velocidad-media (mm/s)		2200 Aprox	2100 Aprox
Repetición de posiciones mm		0.02	
Ancho		19 Aprox	17 Aprox
Recorrido de la Herramienta exp(-2)		Input: 4pts (mano) Output: 4 puntos (base)	
Herramienta de aire		o4x4 Poleas (base-mano)	
Instalación		Estación del piso, aproximación manual	

Figura 3.1.5.3 Especificaciones técnicas Brazo RV-2AJ <sup>33</sup>

<sup>33</sup> Fuente Los Autores

El lenguaje de programación usado en el laboratorio es el MELFA BASIC IV es una versión del lenguaje Basic que ha desarrollado Mitsubishi para sus robots y del que existen las versiones 2, 3 y 4. En la ayuda de CIROS Studio se puede encontrar una extensa información de este lenguaje. El programa MELFA BASIC IV nos presenta un entorno de programación amigable de tal forma que el usuario puede realizar la programación sin complicaciones en forma rápida. Este lenguaje presenta tres instrucciones de movimientos básicos para programar robots industriales.

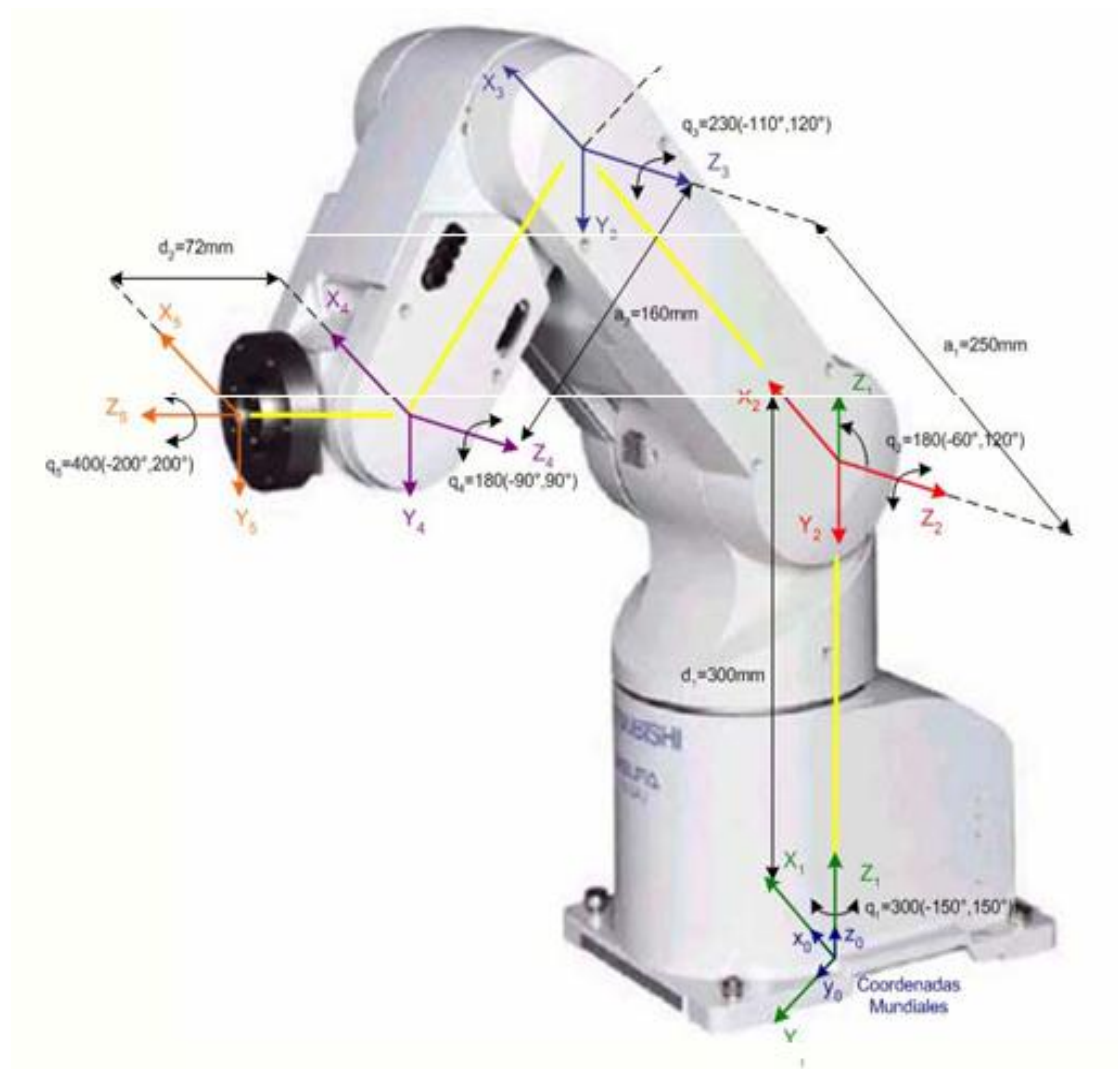


Figura 3.1.5.4 PLANOS DE DESPLAZAMIENTO<sup>34</sup>

<sup>34</sup> Fuente Los Autores

### 3.1.7 ENTORNO DE TRABAJO

En este apartado se hace una descripción un poco detallada de las herramientas utilizadas para el desarrollo de la presente tesis, para que la persona interesada en probar el API, motivo de esta tesis, pueda hacer pruebas exitosas con las herramientas de MITSUBISHI y sepa como guiarse de mejor manera en el desarrollo de su proyecto.

#### 3.1.7.1 COSIROP

COSIROP es un potente entorno de programación, diseñado en 1992 y que se utilizó hasta el 2006, para los robots MITSUBISHI.

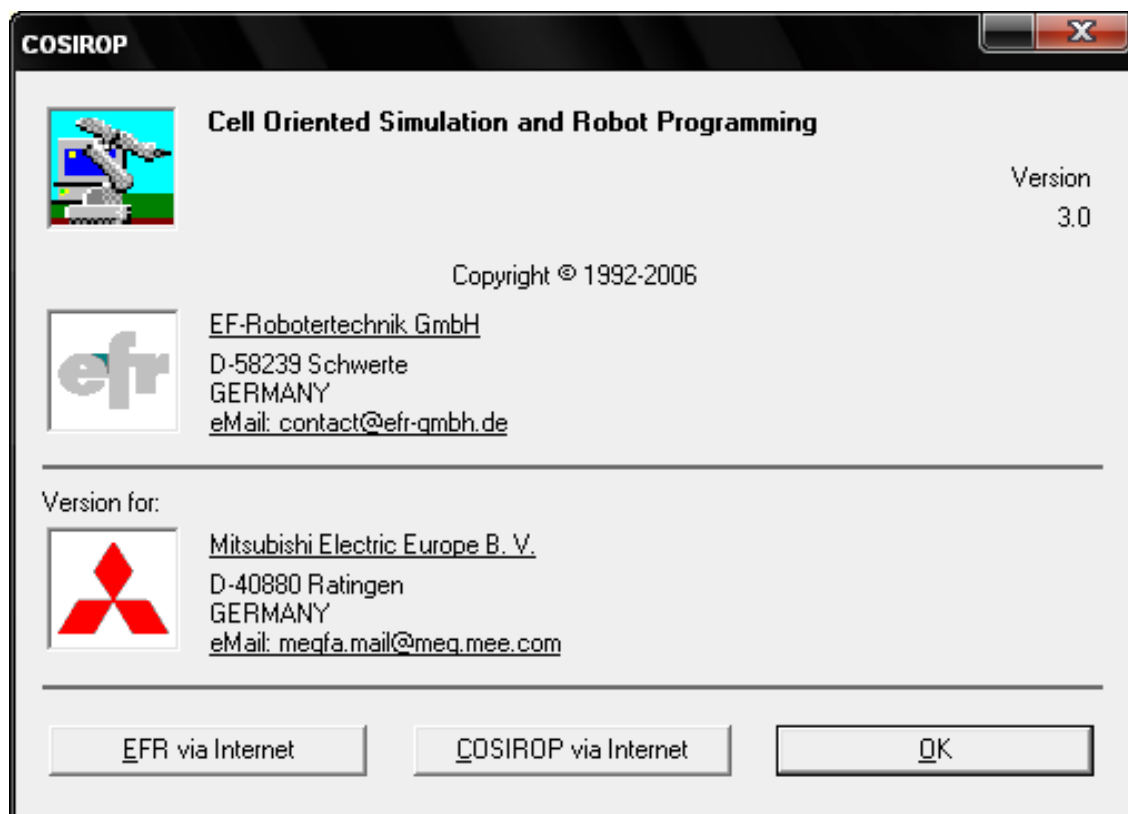


Figura 3.1.7.1.1 INFORMACION SOBRE EL PROGRAMA COSIROP<sup>35</sup>

<sup>35</sup> COSIROP

## CARACTERÍSTICAS

- Es un sistema cerrado y licenciado para controlar brazos robóticos MITSUBISHI.
- El sistema permite crear programas y simulaciones en muy poco tiempo.
- Está diseñado para interactuar directamente con el robot.
- No tiene cargados ejemplos ya que estos vienen en COSIMIR.
- Los lenguajes de programación que utiliza son MELFA BASIC IV o MOVEMASTER COMMAND.
- MELFA BASIC IV es un programa basado en BASIC pero con instrucciones para robots.
- Permite leer programas cargados en la controladora del robot y sus respectivas listas de posiciones
- Permite grabar programas en la controladora y sus respectivas listas de posiciones.
- Mientras el robot está en operación es posible tener una lectura del estado del robot, como por ejemplo :
  - La corriente consumida por sus motores
  - La velocidad de desplazamiento
  - La posición del actuador final a cada momento del movimiento
- Tiene un simulador de Teach que permite controlar el brazo robótico como si se tratara de un Teach físico.
- Permite enviar comandos de prueba desde una consola de comandos.

## INTERFÁZ






Aquí se muestran las diferentes ventanas que tiene el COSIROP para desempeñar sus funciones de programación, control y monitoreo sobre los brazos robóticos MITSUBISHI.

## BARRA DE HERRAMIENTAS






Aquí se muestra una descripción de los botones de la barra de herramientas y su funcionamiento.











Figura 3.1.7.1.2 barra de herramientas<sup>36</sup>

-  **File/New:** permite crear diferentes tipos de programas robot y listas de posición. Se puede elegir entre un nuevo programa robot Movemaster Command (LMR) o un nuevo programa robot MELFA BASIC, o una lista de posiciones LMR. Primero hay que abrir un proyecto / célula de trabajo.
-  **Command File/Open:** permite crear diferentes tipos de programas y listas de posición para el robot. Se puede elegir entre un nuevo programa sea Movemaster Command (LMR), un nuevo programa robot MELFA BASIC o una lista de posiciones.
-  **Command File/Save:** almacena los datos pertenecientes a la ventana activa. Este comando sólo está disponible si la ventana activa contiene un programa de robot, lista de posiciones o muestra una célula de trabajo. Este comando permite guardar modificaciones en un archivo que ya existe.
-  **Command File/Save All:** Almacena todos los programas, las listas de posiciones y la célula de trabajo. Este comando permite guardar las modificaciones a todos los archivos existentes.
-  **Command Edit/Cut:** permite cortar el texto seleccionado desde una ventana y pegar en el portapapeles. Este comando sólo está disponible si la ventana activa contiene una lista de posiciones o un programa de robot y si un texto está seleccionado.

<sup>36</sup> COSIROP

-  **Command Edit/Copy:** permite copiar todo el contenido o las partes seleccionadas (por ejemplo, texto de un programa) de la ventana activa al portapapeles.
-  **Command Edit/Paste:** este comando sólo se activa si la ventana activa contiene un programa de robot o una lista de posiciones.
-  **Command Edit/Renumber:** comando vuelve a numerar las listas de posiciones o programas. Un cuadro de diálogo se abrirá para permitir la entrada de la línea de salida o número de posición y el tamaño del paso.
-  **Command Edit/Find:** permite buscar texto en un programa.
-  **Command Execute/Check Syntax:** permite verificar la corrección sintáctica del programa del robot en la ventana activa. COSIROP comienza el comprobador de sintaxis para el lenguaje de programación de robots específicos. De acuerdo con el robot utilizado la sintaxis del programa puede variar. El comprobador de sintaxis toma en consideración estas diferencias. Los resultados de la comprobación de sintaxis se mostrará en la ventana de mensajes.
-  **Command – RCI Explorer:** abre el RCI Explorer.
-  **Command – Current Error:** despliega una ventana con los errores del robot.
-  **Command Download PC->Robot:** permite transmitir un programa o una lista de posiciones desde el PC hacia el robot. Este comando sólo está disponible si la ventana activa es una ventana de programa de robot o una ventana de lista de posiciones.
-  **Command Upload Robot->PC:** permite transmitir un programa o una lista de la posiciones del robot a la PC. Este comando sólo está disponible si la ventana activa es una ventana de programa robot o una ventana de lista de posiciones.

-  **Command Robot Position->PC:** lee la posición actual del robot de la unidad de disco y coloca el robot simulado en esa posición.
-  **Command Alarm Reset:** Para los robots de la RV-M y la serie RV-E este comando cambia una condición de alarma del robot, enciende los motores del robot y reinicia el programa. Para los robots de la serie A (RV-A, RH-AH, AH y RP) este comando sólo restablece una situación de alarma del robot. Esto también se puede hacer pulsando el botón RESET en el control del robot.
-  **Command Init Connection:** comprueba la conexión entre el PC y el controlador del robot (unidad de disco).
-  **Command Jog Operation:** abre la ventana de Jog Operation para poder manipular el robot desde el Teach virtual.
-  **Command Command tool:** abre la ventana de comandos, que permite probar comandos directamente con el robot.
-  **Command I/O Monitor:** abre la ventana de monitor de entradas y salidas.
-  **Menu Commands:** abre la ventana del administrador de proyectos.
-  **Keys for help:** abre la ventana de ayuda.

## INTERFAZ DE CONTROL

La interfaz de control muestra las ventanas que permiten comunicarse, monitorear y controlar el brazo robótico.

## VENTANA DE MODELAMIENTO

En esta ventana se muestran los movimientos que realiza el robot según la programación.

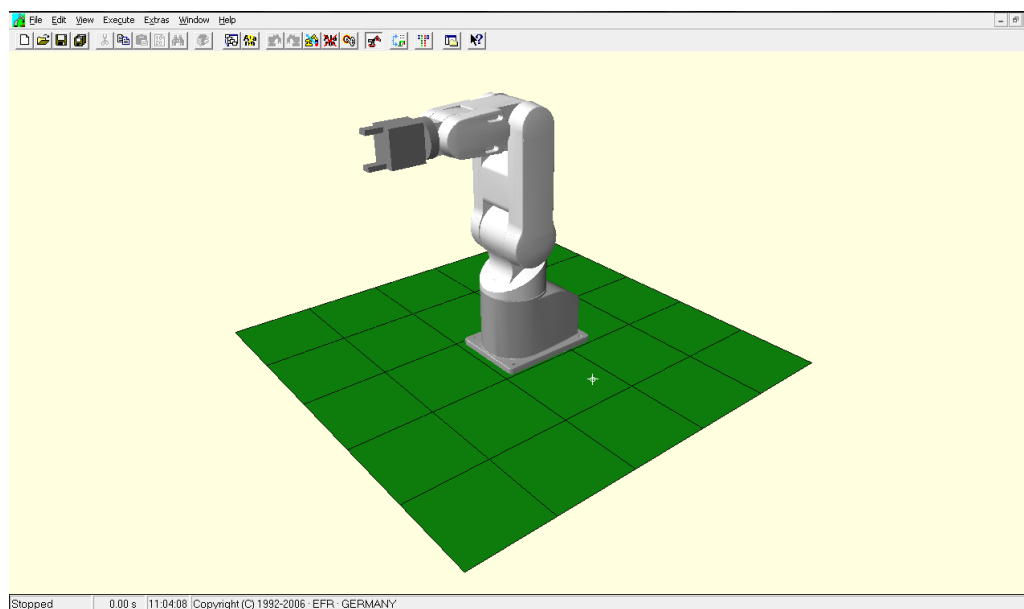


Figura 3.1.7.1.3 ventana de modelamiento<sup>37</sup>

## RCI EXPLORER

El RCI Explorer es un centro de intercambio de información con el brazo robótico. Da una visión general del estado actual del robot y proporciona una manera intuitiva para cargar, descargar, iniciar, depurar y monitorear los programas de robot.

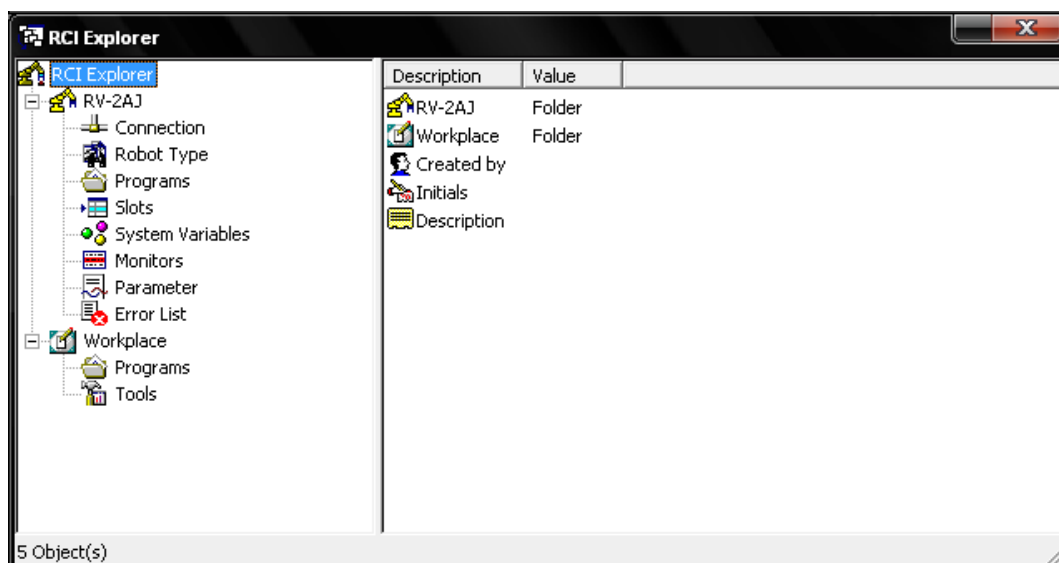


Figura 3.1.7.1.4 ventana de rci explorer<sup>38</sup>

<sup>37</sup> COSIROP

<sup>38</sup> COSIROP



El RCI EXPLORER contiene 2 carpetas.

### CARPETA DE ROBOT

El nombre de la carpeta coincide con el nombre del robot, se ha configurado en el asistente de proyectos. La carpeta de robot permite el acceso a los datos en el controlador del robot. Aquí se encuentra:

- La conexión con el controlador del robot
- El tipo de robot conectado
- Los programas actualmente disponibles en el controlador del robot
- El contenido de las ranuras del programa
- La variables del sistema

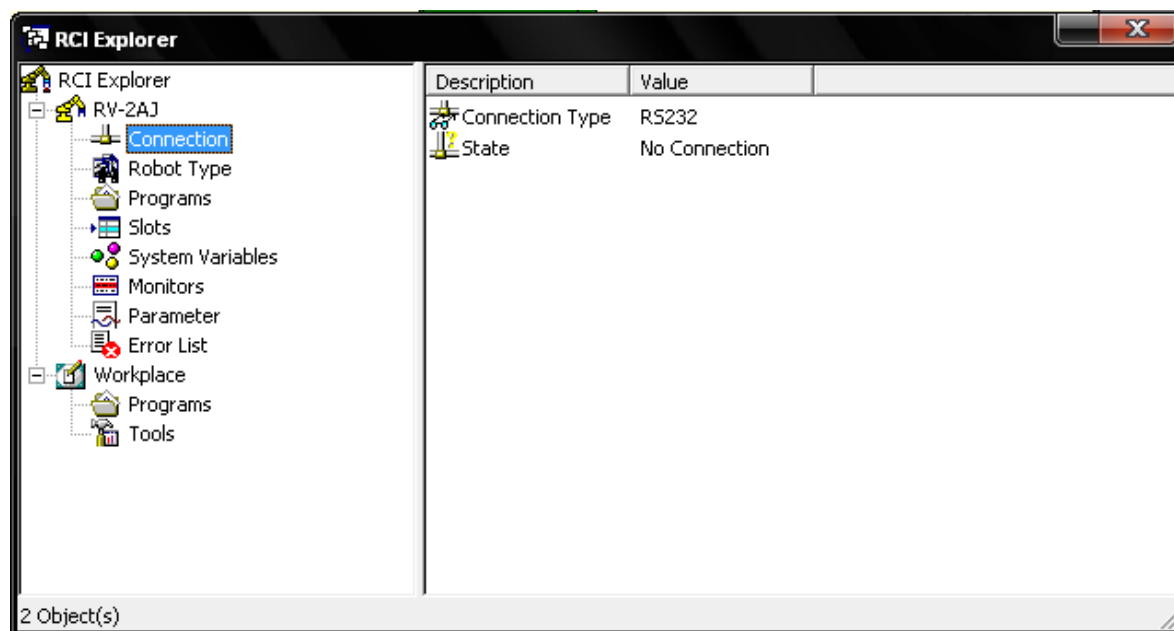


Figura 3.1.7.1.5 ventana de conexión<sup>39</sup>

<sup>39</sup> COSIROP

Si se da doble click sobre cualquiera de los 2 ítems de la derecha se muestra la ventana de configuración para la comunicación serial.

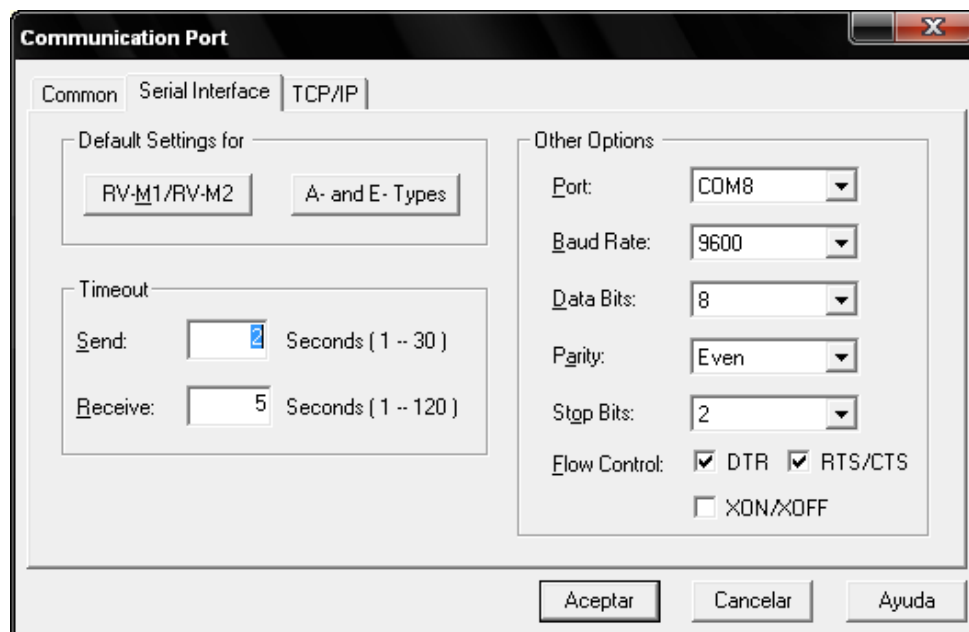


Figura 3.1.7.1.6 ventana de configuración de comunicación serial<sup>40</sup>

Esta ventana permite setear los parámetros para una comunicación efectiva con el robot.

Los parámetros mostrados arriba son los adecuados para comunicarse con el RV-2AJ.

- El estado del robot utilizando una amplia gama de monitores, como por ejemplo:

<sup>40</sup> COSIROP

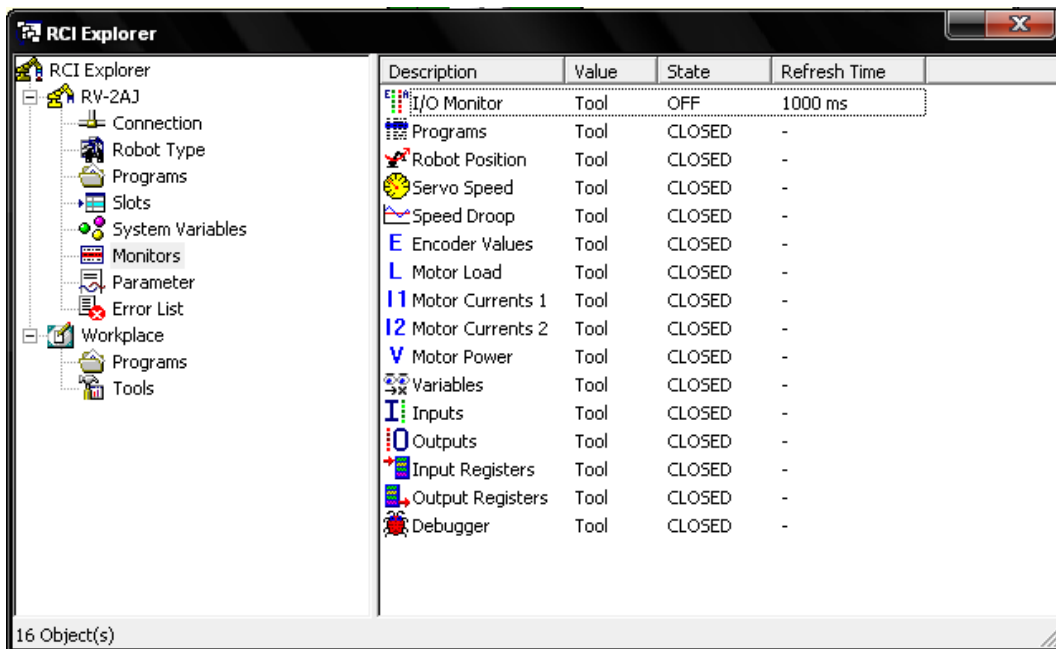


Figura 3.1.7.1.7 ventana de monitores<sup>41</sup>

- **I/O Monitor:** muestra los estados de entrada y salida de la unidad conectada. Permite activar y desactivar las salidas haciendo clic en los indicadores de resultados.

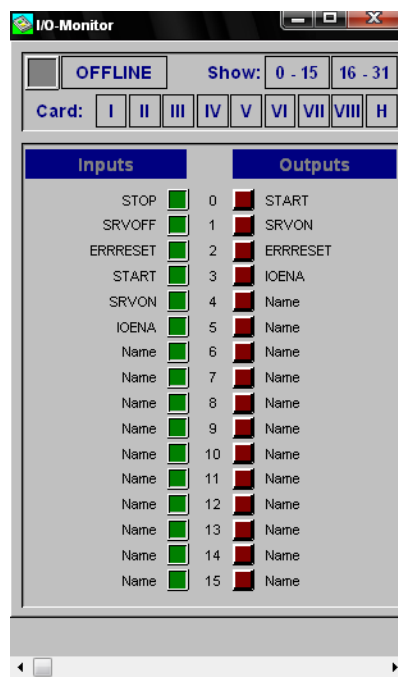


Figura 3.1.7.1.8 ventana de entradas/salidas<sup>42</sup>

<sup>41</sup> COSIROP

- **Program Monitor:** El monitor de programa le permite monitorear el progreso de los programas activos en ranuras diferentes. Se puede cargar como monitor el programa que desee. El monitor de programa permite controlar y depurar programas multitarea con facilidad.

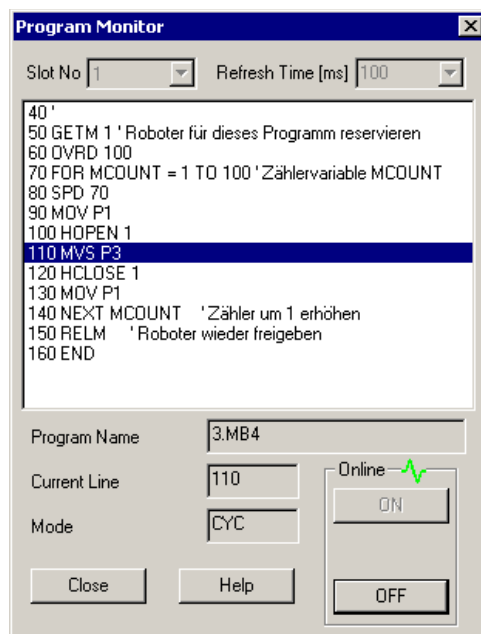


Figura 3.1.7.1.9 ventana monitor de programas<sup>43</sup>

- **Robot Position:** muestra las posiciones reales del brazo robótico mostrando la información por su ángulo de giro y por la posición en el espacio del actuador final. Permite setear la velocidad de consulta al robot, lo normal es a 100 ms.

<sup>42</sup> COSIROP

<sup>43</sup> COSIROP

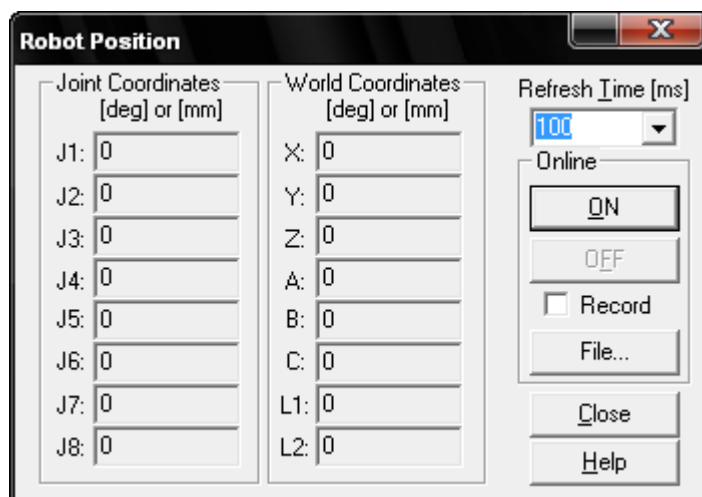


Figura 3.1.7.1.10 ventana de posición del robot<sup>44</sup>

- **Servo Speed:** muestra información cada 100ms, de la velocidad de cada uno de los motores de cada una de las articulaciones del brazo robótico.

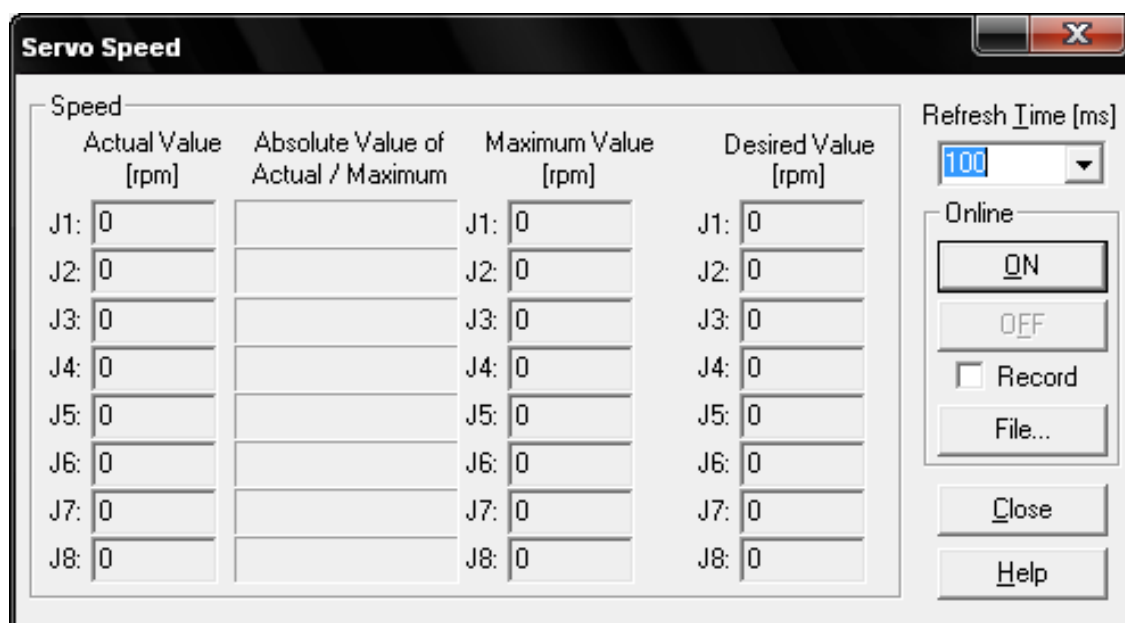


Figura 3.1.7.1.11 ventana de velocidad de servomotores<sup>45</sup>

<sup>44</sup> COSIROP

<sup>45</sup> COSIROP

- **Speed Droop:** muestra los valores máximos y mínimos de las variaciones de velocidad de los motores cada intervalo de tiempo.

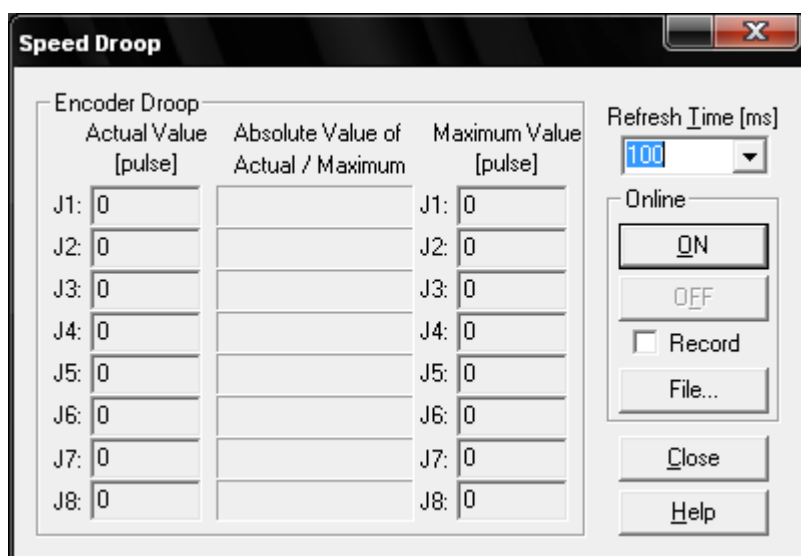


Figura 3.1.7.1.12 ventana de caída de velocidad<sup>46</sup>

- **Encoder Values:** permite controlar los valores actuales de encoder de los servomotores.

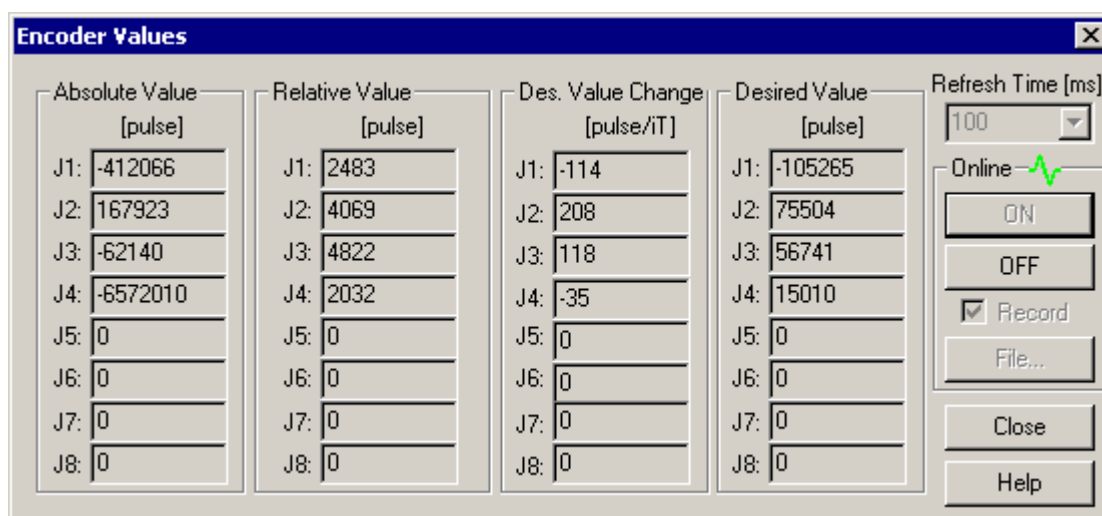


Figura 3.1.7.1.13 ventana de valores de encoders<sup>47</sup>

<sup>46</sup> COSIROP

<sup>47</sup> COSIROP

- **Motor Load:** Permite controlar el nivel de carga de los servomotores. Seleccione un nivel de alerta entre 0 y 100%. Si la carga de uno de los ejes del robot se eleva por encima del nivel de alerta una señal de alerta comenzará a parpadear en la línea correspondiente. La señal de advertencia desaparece si la carga cae por debajo del nivel de alerta.

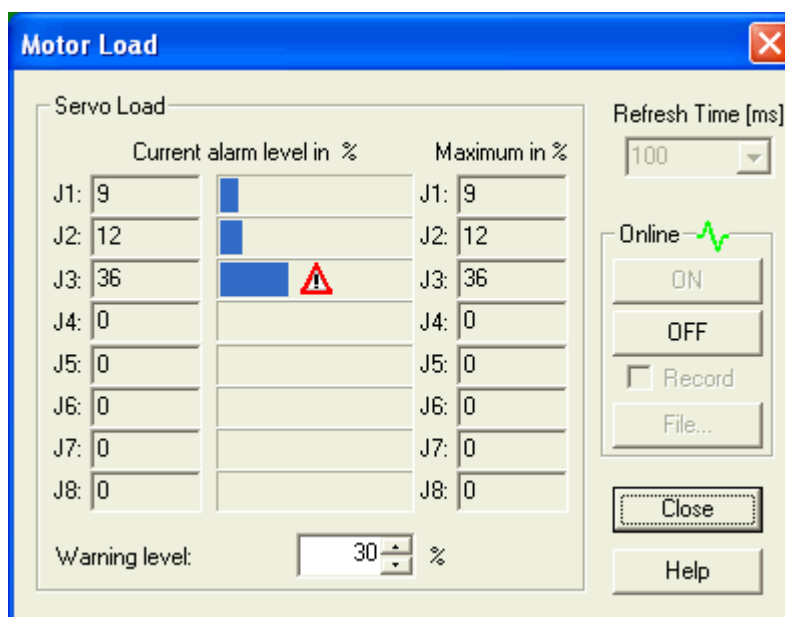


Figura 3.1.7.1.14 ventana de carga de servomotores<sup>48</sup>

- **Motor Currents 1:** permite controlar el nivel máximo de corriente que recibirá cada servomotor. Cuando sobrepase dicho valor una alarma se activará.

<sup>48</sup> COSIROP

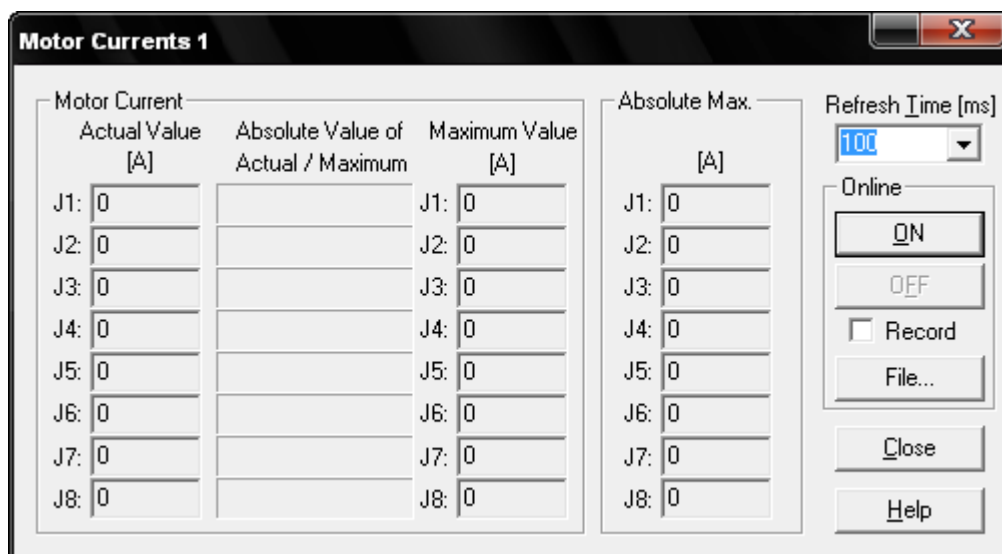


Figura 3.1.7.1.15 ventana de corriente en servomotores<sup>49</sup>

- **Motor Currents 2:** permite controlar el valor mínimo y máximo de corriente en cada motor. Una vez que supere por arriba o por abajo a los valores seteados una alarma se activará.

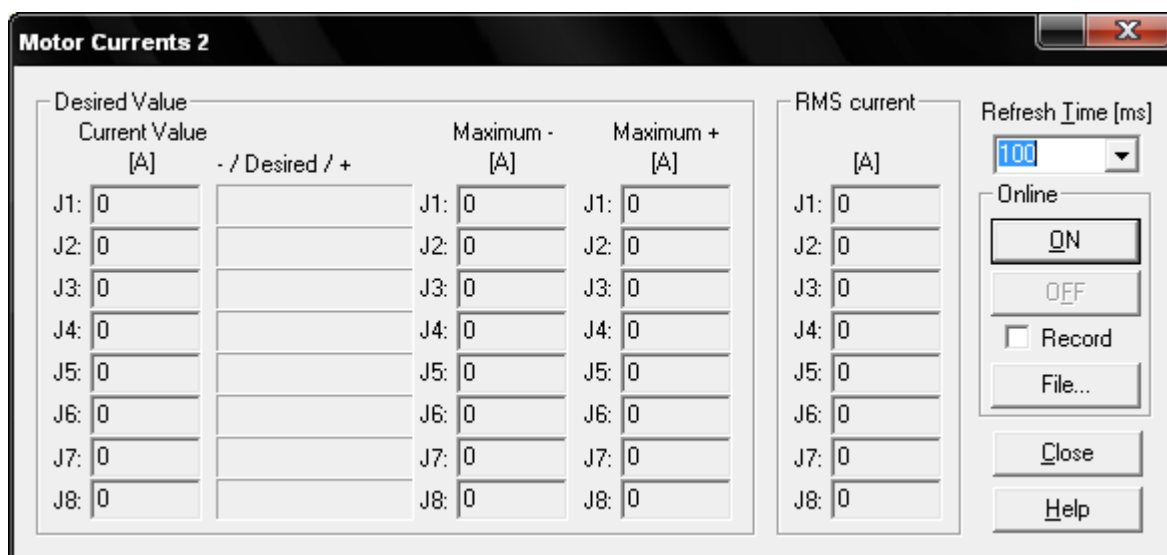


Figura 3.1.7.1.16 ventana de corriente en servomotores<sup>50</sup>

<sup>49</sup> COSIROP

<sup>50</sup> COSIROP



- **Motor Power:** muestra la potencia consumida en cada servomotor.

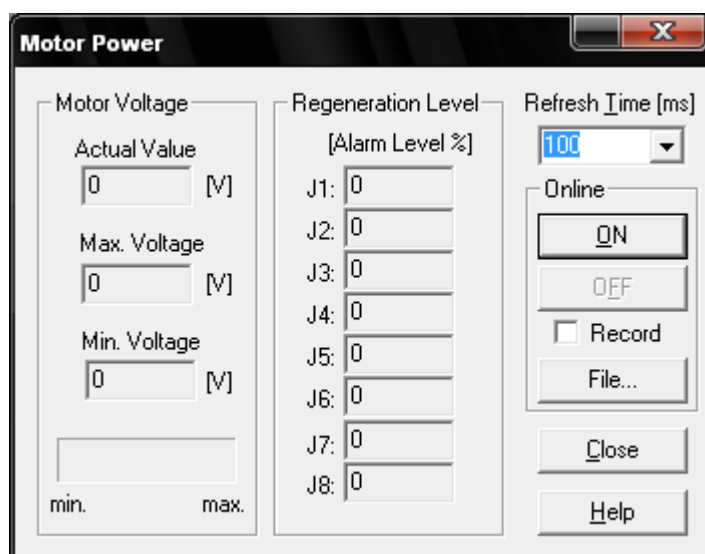


Figura 3.1.7.1.17 ventana de potencia de servomotores<sup>51</sup>

- **Variables Monitor:** permite controlar las variables seleccionadas en función del ciclo. Se pueden abrir tantos monitores como se desee. La lista de la derecha muestra las variables con los valores y en la lista de la izquierda se pueden ingresar las variables deseadas.

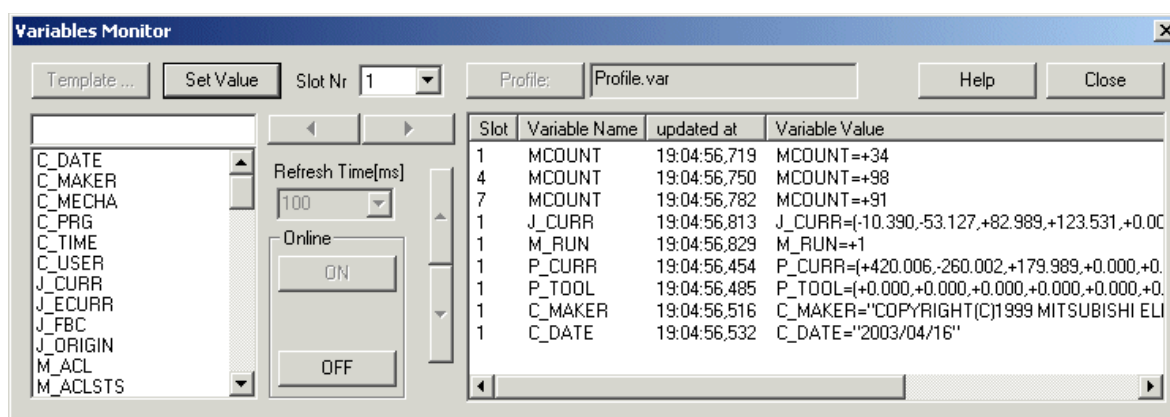


Figura 3.1.7.1.18 ventana de monitor de variables<sup>52</sup>

<sup>51</sup> COSIROP

<sup>52</sup> COSIROP

- **Inputs:** Cada monitor de entrada puede controlar 16 valores sucesivos de entrada binaria, al mismo tiempo. Los datos mostrados se actualizarán de acuerdo con el tiempo de actualización dado. Los valores de entrada binaria se muestran como números y luces verdes (verde oscuro = 0, verde claro = 1).

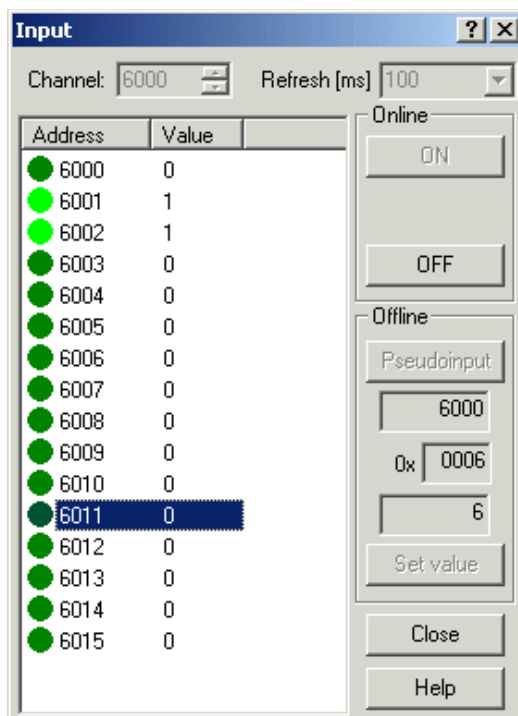


Figura 3.1.7.1.19 ventana de entradas<sup>53</sup>

- **Outputs:** Cada salida de monitor puede controlar 16 valores sucesivos de salida binaria, al mismo tiempo. Los datos mostrados se actualizarán de acuerdo con el tiempo de actualización dado. Los valores de salida binaria se muestran como números y luces rojas (rojo oscuro = 0, rojo claro = 1).

<sup>53</sup> COSIROP

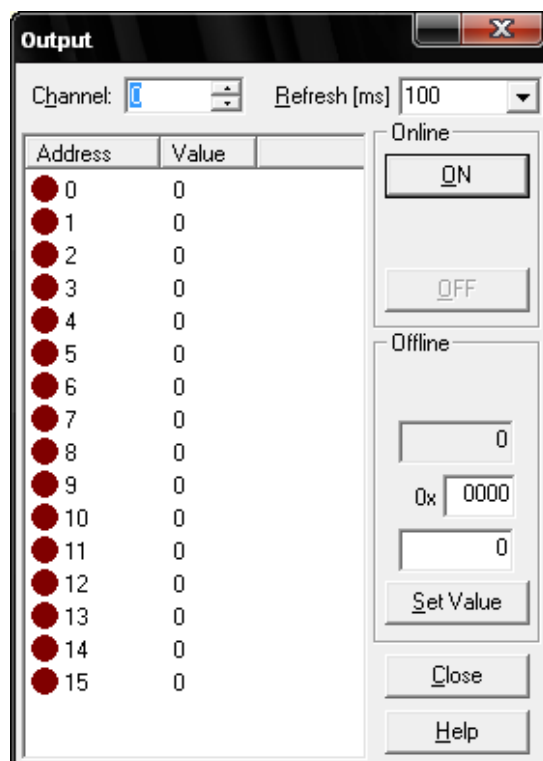


Figura 3.1.7.1.20 ventana de salidas<sup>54</sup>

- **Input Register Monitor:** Cada registro de entrada del monitor puede controlar 16 registros de entradas sucesivas al mismo tiempo. Los datos mostrados se actualizarán de acuerdo con el tiempo de actualización dado. Los valores de los 16 registros de entrada se muestra como decimal y como números hexadecimales.

<sup>54</sup> COSIROP

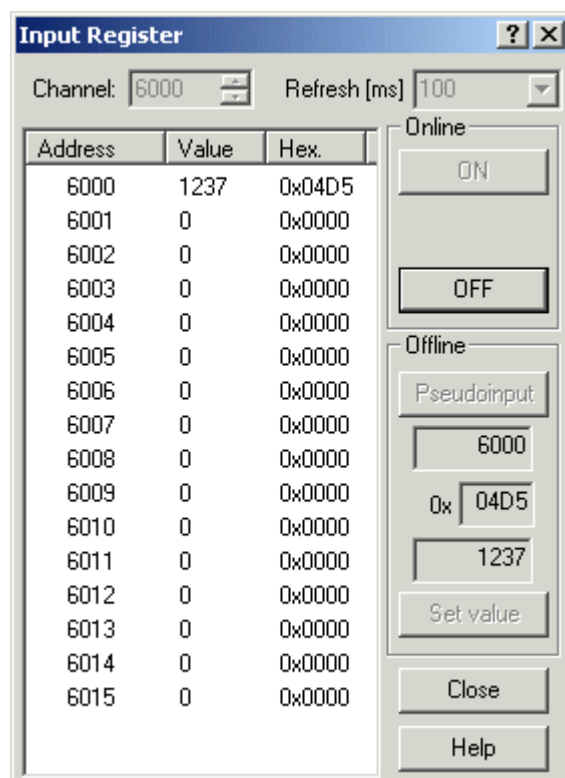


Figura 3.1.7.1.21 ventana de registro de entradas<sup>55</sup>

- **Output Register Monitor:** Cada registro de salida del monitor puede controlar 16 registros de salidas sucesivas al mismo tiempo. Los datos mostrados se actualizarán de acuerdo con el tiempo de actualización dado. Los valores de los 16 registros de salida se muestra como decimal y como números hexadecimales.

<sup>55</sup> COSIROP

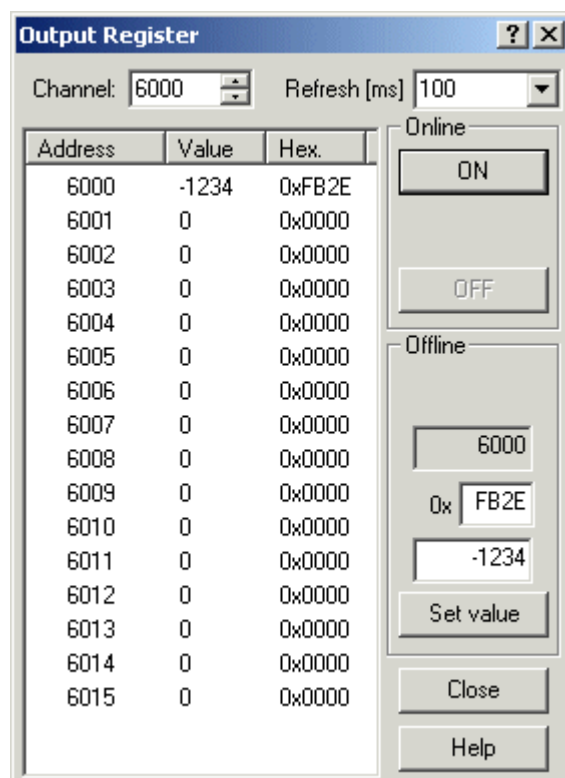


Figura 3.1.7.1.22 ventana de registro de salidas<sup>56</sup>

- **Debugger:** El depurador le permite comprobar directamente la funcionalidad de los programas individuales. Primero, cargar un programa. Luego, añadir puntos de interrupción en su programa. La ejecución del programa se detendrá en un punto de interrupción antes de ejecutar la línea de programa que contiene el punto de interrupción. Se establece un punto haciendo doble clic sobre la línea del programa o presionando el icono Set/Reset Breakpoint. Para borrar un punto de interrupción, haga doble clic de nuevo en la línea de programa o pulse de nuevo icono Set/Reset Breakpoint. Un icono rojo de STOP delante de la línea de programa significa un punto de interrupción.

<sup>56</sup> COSIROP

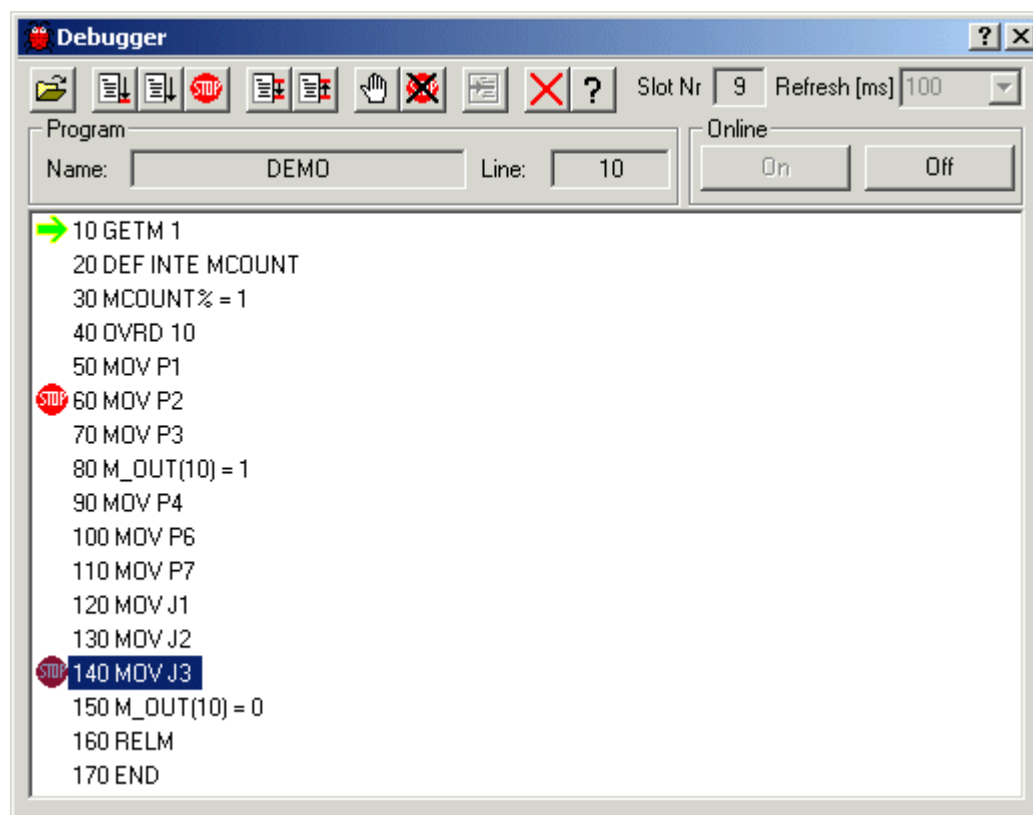


Figura 3.1.7.1.23 ventana del debugger<sup>57</sup>

## LUGAR DE TRABAJO

Contiene los datos de su PC en el directorio del proyecto y algunas herramientas para controlar el brazo robot.

- **Programas:** muestra un listado de los programas y lista de posiciones almacenados en memoria. Se puede revisar los programas guardados dando doble click sobre ellos.

<sup>57</sup> COSIROP

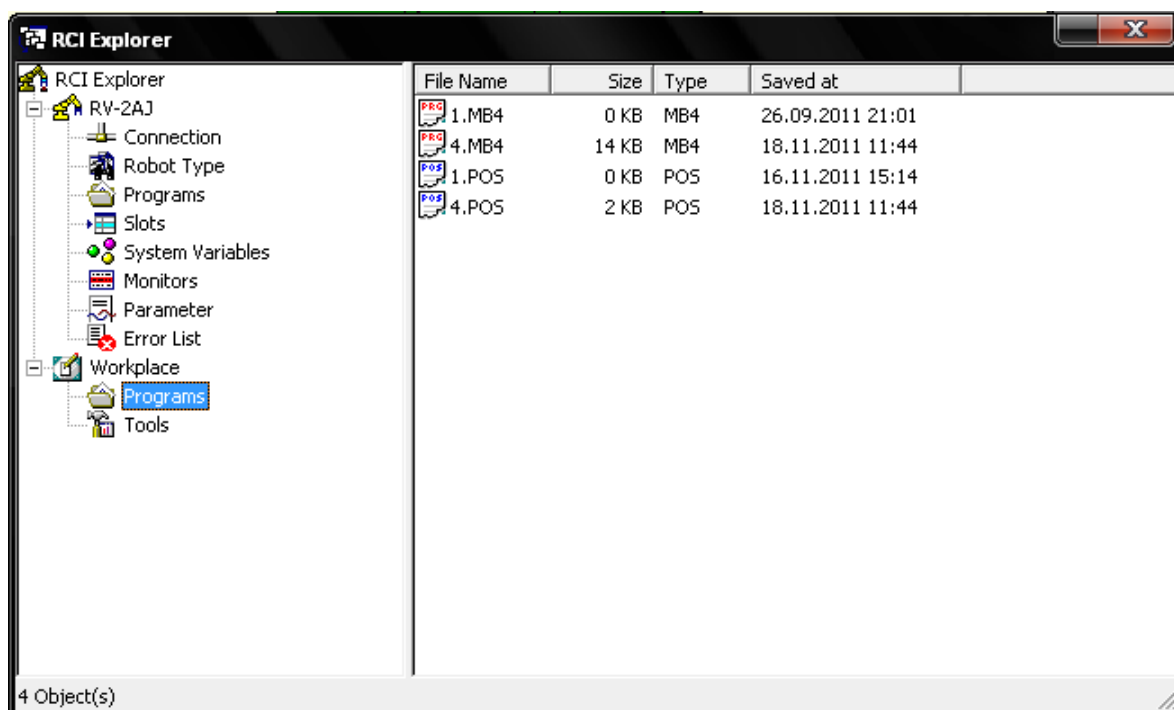


Figura 3.1.7.1.24 ventana RCI Explorer<sup>58</sup>

- **Tools:** contiene un grupo de herramientas que permiten controlar el brazo robótico.

<sup>58</sup> COSIROP

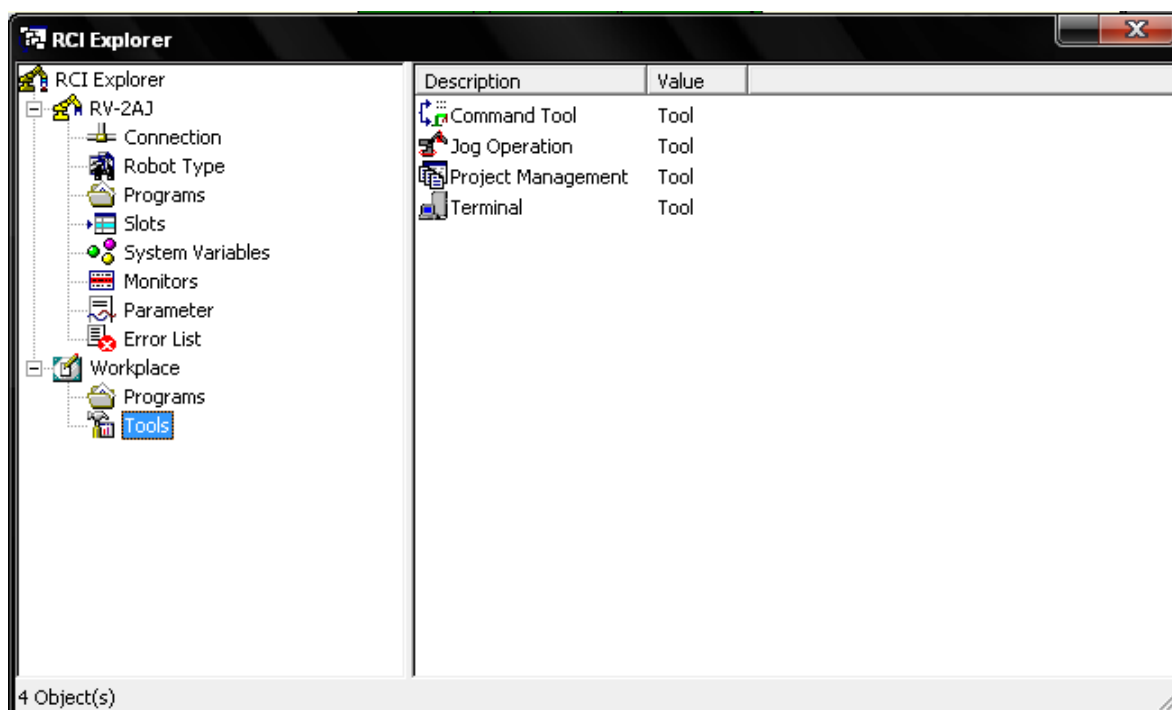


Figura 3.1.7.1.25 ventana de herramientas de control<sup>59</sup>

- **Command Tool:** muestra todos los comandos que son compatibles con su robot. Los comandos del robot se dividen en siete grupos de comandos. La barra de herramientas de comandos contiene un símbolo para cada grupo. Para ver todos los comandos en orden alfabético, seleccione el símbolo correspondiente en la barra de herramientas.

<sup>59</sup> COSIROP



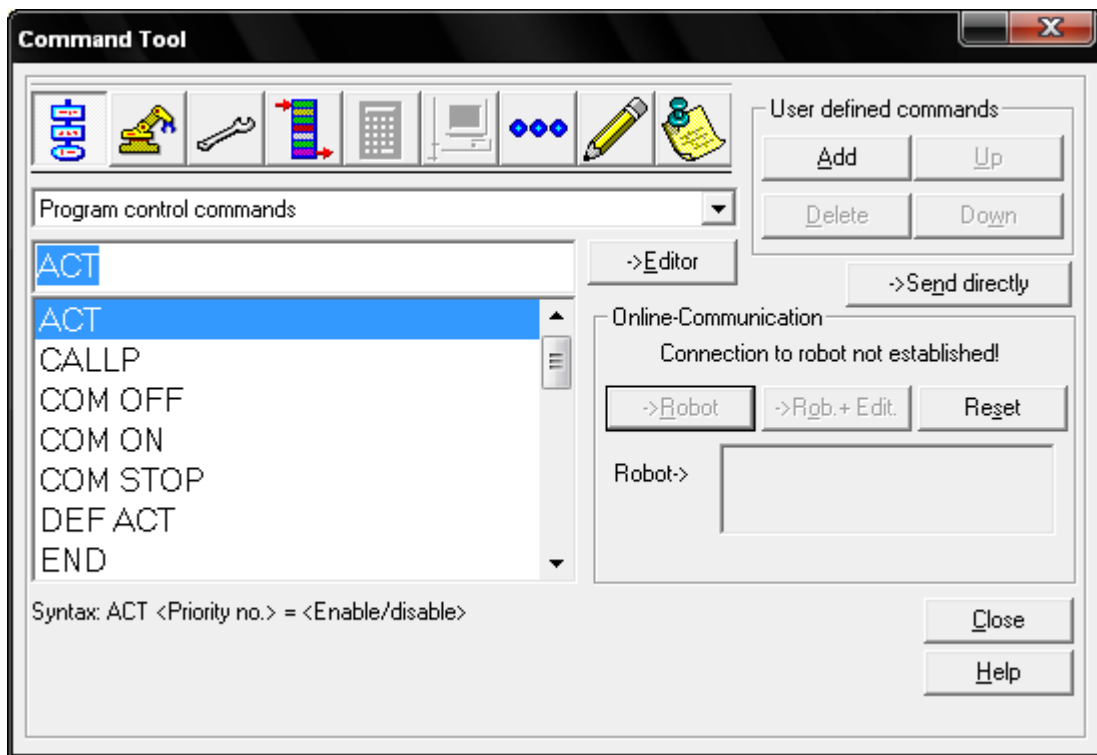






Figura 3.1.7.1.26 ventana de comandos<sup>60</sup>

-  **Program Control Commands:** muestra comandos de programa.
-  **Position and Motion Control:** muestra comandos de posicionamiento y movimiento.
-  **Hand Control Commands:** muestra comandos de control de la mano.
-  **I/O Control Commands:** muestra comandos de entrada y salida.

<sup>60</sup> COSIROP

-  **Operation, substitute, exchange commands:** muestra comandos de intercambio, operación y sustitución con la controladora del robot.
  -  **RS232C communication commands:** muestra comandos para probar la comunicación serial con el robot.
  -  **Other commands:** muestra comandos que no entran en los demás grupos.
  -  **User defined commands:** muestra comandos definidos por el usuario.
  -  **Last commands:** muestra los últimos comandos utilizados.
- **Jog Operation:** es un teach virtual que permite mover el brazo como si fuera un teach físico. En esta ventana podemos controlar la apertura y cierre de la mano y la posición en XYZAB. El botón “SET XYZ POSITION”, permite setear una posición que nosotros queramos para la herramienta en función de XYZAB.

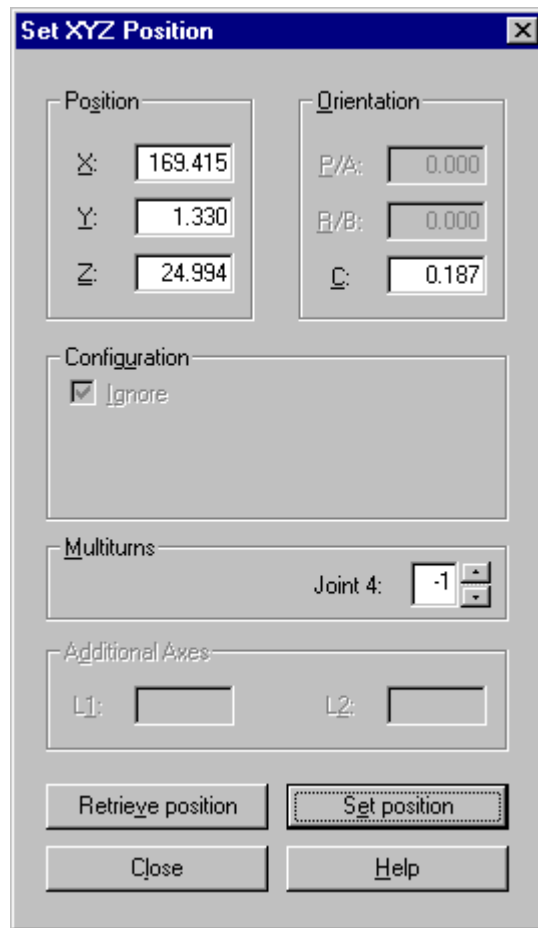


Figura 3.1.7.1.27 ventana de posicionamiento en XYZ<sup>61</sup>

Si presionamos Set Position el robot se sitúa en la posición establecida, pero si presionamos Retrieve Position, obtenemos la posición actual del robot en el mundo real en XYZAB.

En Jog Operation encontramos 3 modos de operación:

- **XYZ Jog:** el robot se mueve en las coordenadas del mundo (XYZAB) por interpolación lineal. Las flechas en los botones indican la dirección de desplazamiento positivo o negativo para los movimientos de traslación a lo largo de los ejes. Con el fin de determinar la dirección positiva o negativa para los movimientos rotatorios se aplica la "regla de la mano derecha".

<sup>61</sup> COSIROP

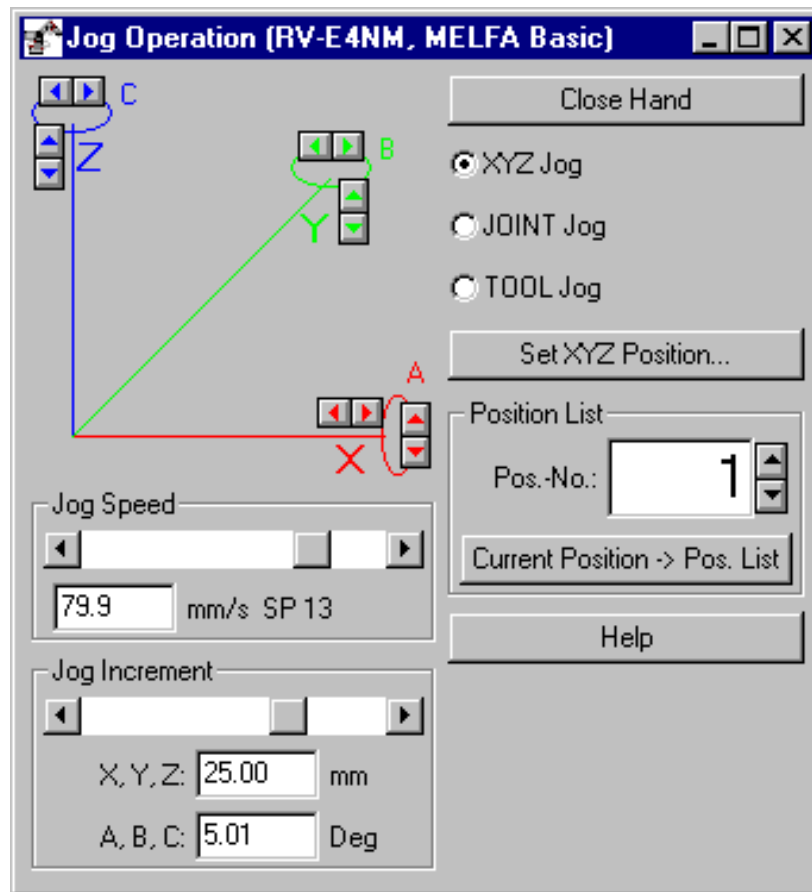


Figura 3.1.7.1.28 ventana de movimiento por XYZ<sup>62</sup>

- **JOINT Jog:** en este modo, el robot se mueve por separado y por interpolación de sus articulaciones. La flecha en el botón apuntando a la izquierda, indica movimiento negativo y la flecha apuntando hacia la derecha, indica movimiento positivo.

<sup>62</sup> COSIROP

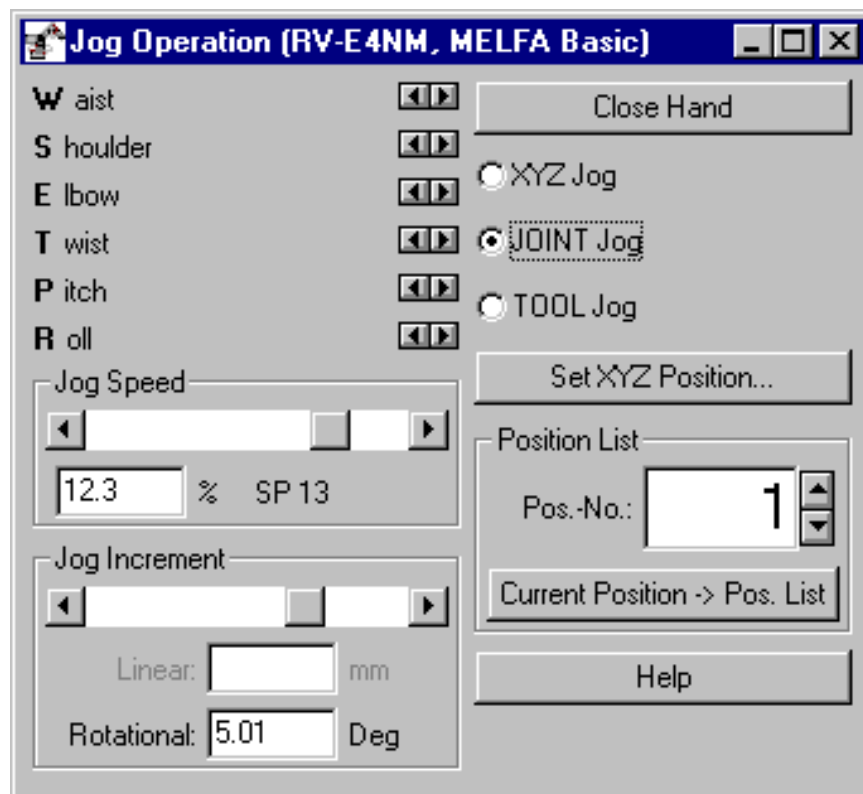


Figura 3.1.7.1.29 ventana de movimiento por articulación<sup>63</sup>

- **TOOL Jog:** el robot se mueve en las coordenadas de la herramienta por interpolación lineal, tomando como eje 0 el centro de la herramienta. Las flechas en los botones indican la dirección de desplazamiento positivo o negativo para los movimientos de traslación a lo largo de los ejes. Con el fin de determinar la dirección positiva o negativa para los movimientos rotatorios se aplica la "regla de la mano derecha".

<sup>63</sup> COSIROP

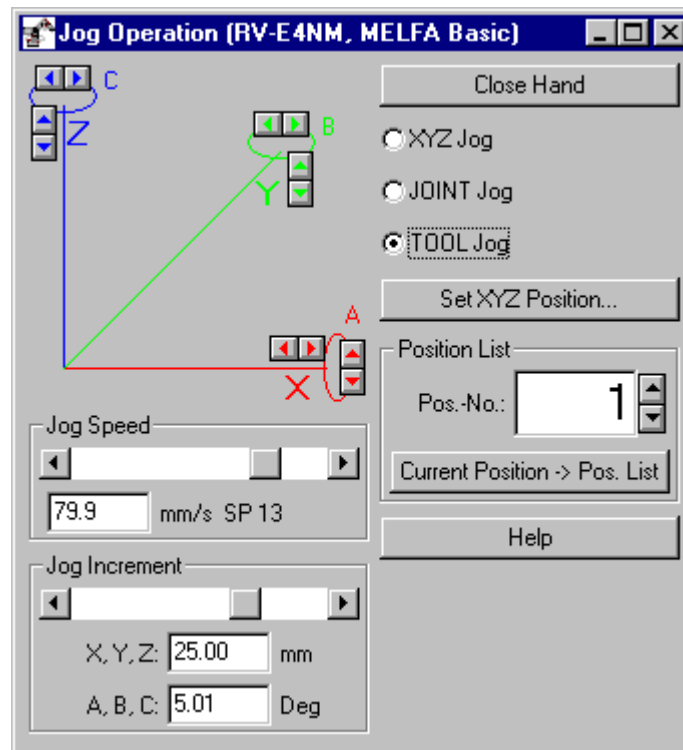


Figura 3.1.7.1.30 ventana de movimiento por pinza<sup>64</sup>

- **Project Management:** esta ventana nos permite definir el proyecto principal con el que estaremos trabajando en la simulación. Nos da la opción de explorar los diferentes proyectos que tengamos almacenados dentro de la computadora, compilarlos y definirlos como proyectos activos dentro de nuestro trabajo. Nos muestra información del sistema de archivos del programa, sea el archivo de código como el archivo de posiciones.

<sup>64</sup> COSIROP

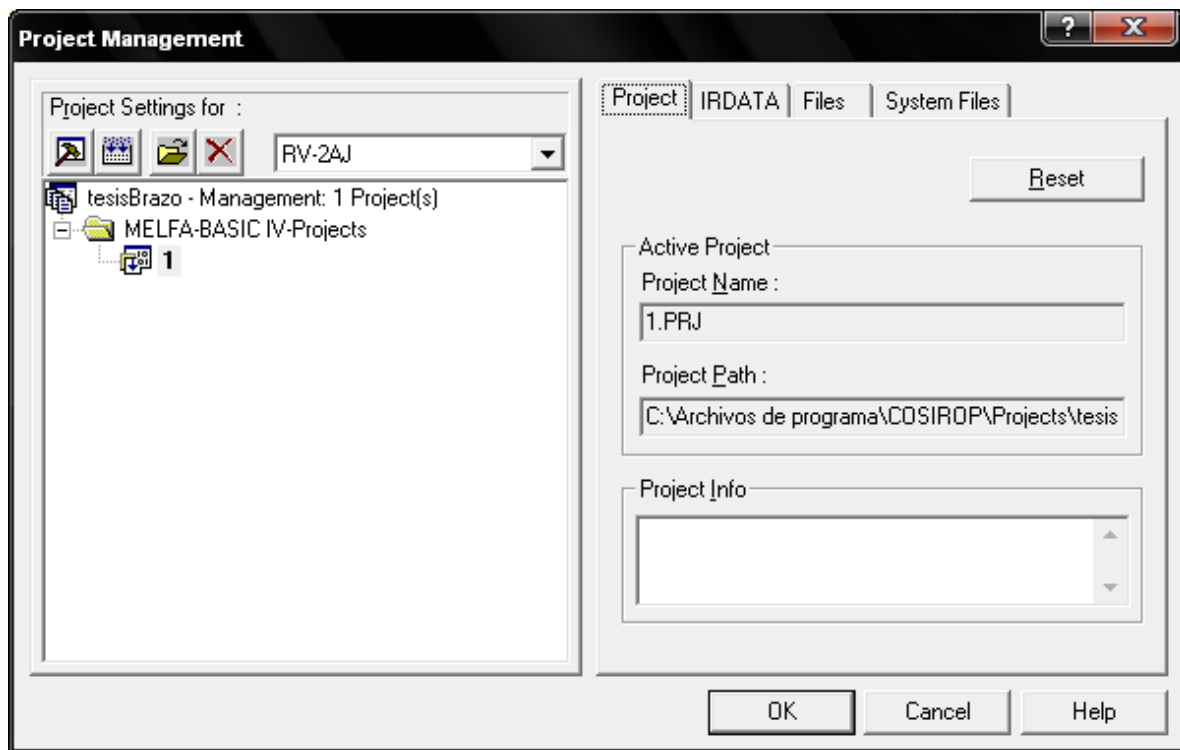


Figura 3.1.7.1.31 ventana de manejo de proyectos<sup>65</sup>

- **Terminal:** se utiliza para comunicarse directamente con el robot. Insertar cualquier mensaje y transmitirlo al robot, simplemente pulsando la tecla <ENTER> en el teclado.

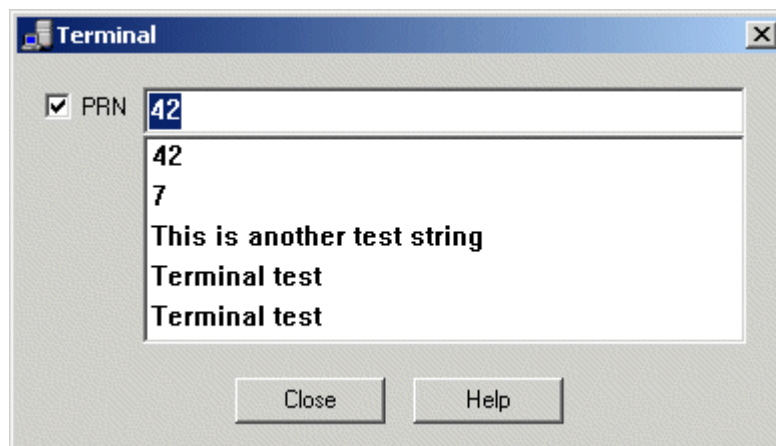


Figura 3.1.7.1.32 ventana de terminal de comunicación<sup>66</sup>

<sup>65</sup> COSIROP

<sup>66</sup> COSIROP

## INTERFAZ DE PROGRAMACIÓN

La interfaz de programación está compuesta de 3 ventanas principales.

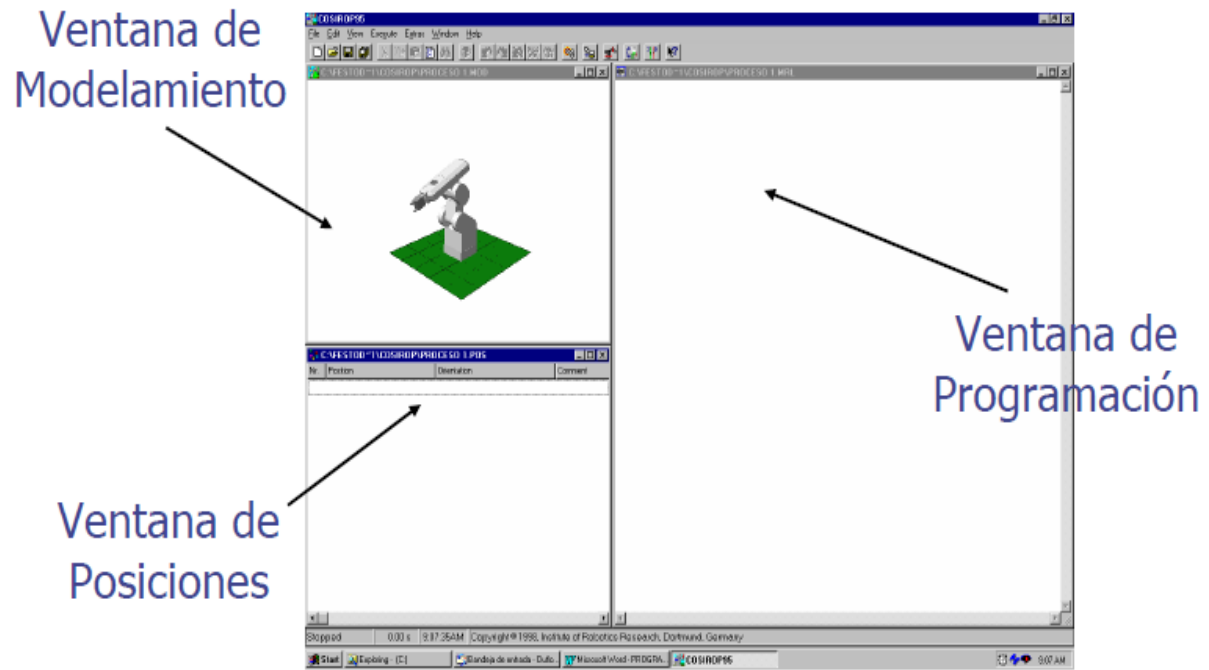


Figura 3.1.7.1.33 ventana de programación<sup>67</sup>

Fuente: DIPLOMADO INTERNACIONAL EN AUTOMATIZACION DE PROCESOS INDUSTRIALES, Programación del Robot Mitsubishi con el software COSIROP

- **Ventana de posiciones:** es el lugar donde almacenamos las posiciones a las que se moverá el robot. Se puede agregar estas posiciones de manera manual, o directamente del robot simulado o el real.

<sup>67</sup> COSIROP



No	Position	Orientation	Comment	
VECZ2	0.0, 0.0, 2.5	0, 0		
VECZ2	0.0, 0.0, 20.0	0, 0		
VECZ3	0.0, 0.0, 30.0	0, 0		
VECZ5	0.0, 0.0, 50.0	0, 0		
VECZ2	0.0, 0.0, 200.0	0, 0		
VECZINT	0.0, 0.0, 0.5	0, 0		
VECGA	40.0, 0.0, -2.0	0, 0		
AUXPO	-65.7, 190.9, 384.0	-90, 180, 0, 0, R, A		
VECMII	0.0, 0.0, 0.0	1, 0		
VECCA	0.0, 0.0, 0.0	0, 0, 0, 0		
VECCU	16.3, -30.3, -8.0	0, 0		
VECCM	56.3, -76.3, -25.0	0, 0		
VECCO	-0.3, 0.0, -3.5	0, 0		
VECCO	15.0, -76.3, -19.0	0, 0		
P1	251.7, 53.3, 165.8	-52, 180, 0, 0, R, A		
P2	218.1, 203.3, 163.6	0, 180, 0, 0, R, A		
P3	187.7, 219.7, 141.4	1, 181, 0, 0, R, A		
P4	82.6, 193.1, 334.1	-90, 180, 0, 0, R, A		

Figura 3.1.7.1.34 ventana de lista de posiciones<sup>68</sup>

- **Ventana de programación:** es el lugar donde se escribe el código del programa que queremos realizar. Aquí se realiza la verificación de la sintaxis. El programa siempre tiene que ir numerado antes de la compilación, y la numeración se realiza de 10 en 10, por si se necesita agregar alguna línea de código intermedia, no haya que reenumerar todo otra vez.

```

850 'correction orientation measurement - assembly position cylinder
860 '
870 VECZ2 = (+0.00,+0.00,+2.50,+0.00,+0.00) 'correction z-axis 2.5 mm / Korrektur Z-Achse 2.
880 VECZ20 = (+0.00,+0.00,+20.00,+0.00,+0.00) 'correction z-axis 20 mm / Korrektur Z-Achse 20
890 VECZ30 = (+0.00,+0.00,+30.00,+0.00,+0.00) 'correction z-axis 30 mm / Korrektur Z-Achse
900 VECZ50 = (+0.00,+0.00,+50.00,+0.00,+0.00) 'correction z-axis 50 mm / Korrektur Z-Achse 50
910 VECZ200 = (+0.00,+0.00,+200.00,+0.00,+0.00) 'correction z-axis 200 mm / Korrektur Z-Achse 2
920 VECCUP = (+16.25,-30.25,-8.00,+0.00,+0.00) 'Korrektur Teachposition - Umgreifposition
930 'correction teachposition regripp position
940 VECCMPZ = (+56.25,-76.25,-25.00,+0.00,+0.00) 'Korrektur Teachposition - Montageposition Zyl
950 'correction teach position - assembly position cylinder
960 VECCOM = (+15.00,-76.25,-19.00,+0.00,+0.00) 'Korrektur Orientierungsmessung - Montageposit
970 'correction orientation measurement - assembly position cylinder
980 VECCOZ = (-0.25,+0.00,-3.50,+0.00,+0.00) 'Korrektur Teachposition - Orientierung Zylinder
990 'correction teachposition - orientation cylinder position
1000 VECGA = (+40.00,+0.00,-2.00,+0.00,+0.00) 'Korrektur Montageposition - Aussengreifpositi
1010 'correction assembly position - regripp position
1020 VECZINT = (+0.00,+0.00,+0.50,+0.00,+0.00) 'z intervall of measurement spring present /
1030 'Z Intervall Federvorhandenmessung
1040 VECMINC = (+0.00,+0.00,+0.00,+1.00,+0.00) 'measurement intervall orientation meassuremer
1050 'Messintervall Orientierungsmessung
1060 '
1070 ' definition of external io adresses / Definition der externen EA Adressen
1080 DEF IO PART_AV = BIT,8 'Werkstück vorhanden / Workpiece available

```

Figura 3.1.7.1.35 ventana de programación<sup>69</sup><sup>68</sup> COSIROP<sup>70</sup> COSIROP

- **Ventana de modelamiento:** muestra los movimientos del robot que están en la ventana de programa.

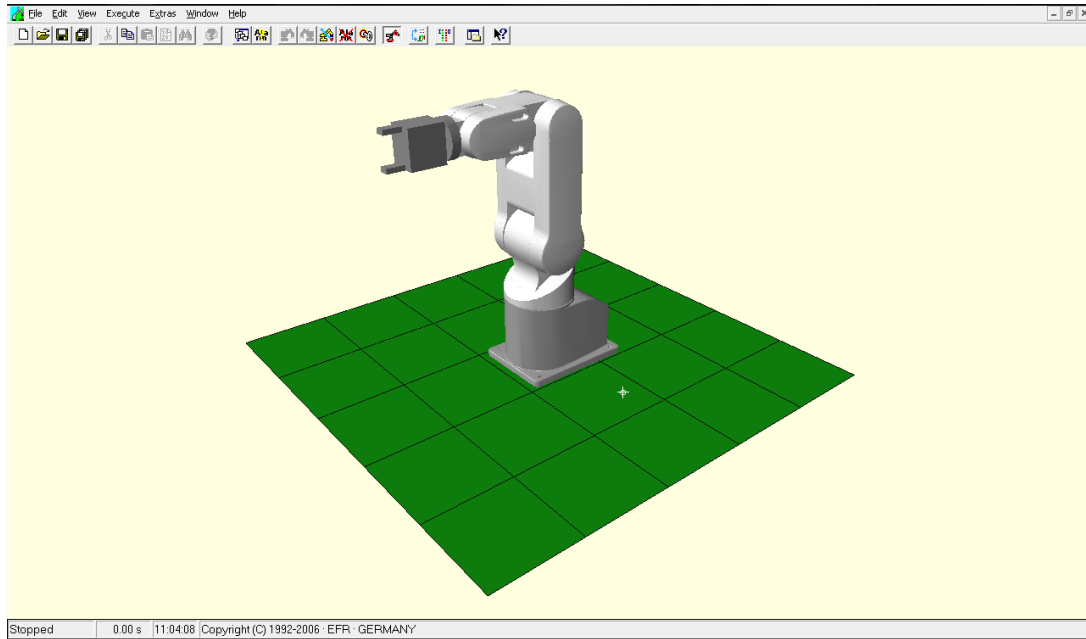


Figura 3.1.7.1.36 ventana de modelamiento<sup>70</sup>

## LENGUAJES DE PROGRAMACIÓN

COSIROP soporta varios lenguajes de programación como son MOVEMASTER COMMAND, MELFA BASIC III Y MELFA BASIC IV.

El siguiente capítulo está orientado a la descripción de MELFA BASIC IV, ya que es el lenguaje que se utilizó para el desarrollo del proyecto y las pruebas a lo largo de la tesis.

## LENGUAJE MELFA BASIC IV

MELFA BASIC es un lenguaje de programación de alto nivel con bases de BASIC<sup>71</sup>, que permite programar los robots de MITSUBISHI de una manera fácil y rápida, sin

---

<sup>70</sup> COSIROP

necesidad de una preparación extensa por parte de los usuarios, sean estudiantes o profesores.

## CARACTERÍSTICAS

- **Coma (,):** sirve como separador cuando se especifican muchos parámetros consecutivos.

**Ejemplo:**


100 P50 = (450,100,300, ...)



- **Punto (.):** sirve para datos múltiples, como los datos posicionales, el punto sirve como separador de cada componente singular.

**Ejemplo:**

110 M10 = P10.X



- Los nombres de variables de tipo posición, articulación (joint), aritmética y cadena de caracteres, empiezan con un carácter particular, como se muestra en la siguiente tabla.

Carácter	Significado
P	Positional (variable de posición)
J	Joint (articulación)
M	Aritméticas o Matemáticas
C	Character String (cadena de caracteres)

Tabla 3.1.7.1.1 Caracteres especiales para posicionamiento<sup>72</sup>

<sup>71</sup> Beginner's All-purpose Symbolic Instruction Code

<sup>72</sup> COSIROP

**Ejemplo:**

<b>Posición P</b>	<b>Articulación J</b>	<b>Matemático M</b>	<b>Caracteres C</b>
P1	J1	M10	C30S
P99	J3	M99	C10S
P1Z	J6	M5	C5S

Tabla 3.1.7.1.2 Caracteres especiales para posicionamiento<sup>73</sup>

Dentro los tipos de instrucciones se pueden encontrar algunas categorías como son:

- **INSTRUCCIONES DE MOVIMIENTO**

Este grupo de instrucciones mueven al robot a un punto específico mediante interpolaciones lineales o cilíndricas. Las trayectorias descritas pueden ser lineales o circulares dependiendo del comando. Las trayectorias para los desplazamientos son calculados por la controladora, determinando siempre el camino óptimo, por lo que el usuario no tiene control absoluto sobre el robot.

El grupo de instrucciones de movimiento se describen en la siguiente tabla.

<b>Comando</b>	<b>Función</b>
ACCEL	Designar a la aceleración del robot y la velocidad de desaceleración como un porcentaje (%).
BASE	Designa la base para conversión de datos.
CMP JNT	Es posible controlar cada uno de los ejes del robot en el sistema de coordenadas común de una manera flexible.
CMP POS	Designa el eje para mover suavemente el robot en el sistema de

<sup>73</sup> COSIROP

	coordenadas ortogonales.
CMP TOOL	Designa el eje para mover suavemente el robot en el sistema de coordenadas de la herramienta.
CMP OFF	Desactiva el modo de control suave.
CMPG	Define la suavidad de movimiento del robot.
CNT	Designa el control de operación continua de las operaciones de interpolación.
COLCHK	Permite activar / desactivar la función de detección de impactos.
COLLVL	Ajusta el nivel de detección de la función de detección de impactos.
DEF JNT	Declara variables de articulación.
DEF POS	Declara variables de posición.
DLY	Cuando se utiliza como un solo comando: produce una espera. Cuando se utiliza como salida de impulsos: designa el tiempo de salida de un pulso.
FINE	Esta instrucción establece las condiciones de la terminación de la posición del robot. No es válido durante el control de movimiento suave.
JOVRD	Designa el remplazo que es válido sólo durante los movimientos por articulación.
JRC	Esta instrucción rescribe las coordenadas actuales mediante la adición de + / -360 grados a los valores de coordenadas actual articulación del eje especificado.
MOV	Usando la operación de interpolación por articulación, se mueve desde la posición actual a la posición de destino.
MVA	Esta instrucción mueve el robot desde la posición actual a la posición de destino con un movimiento de arco.
MVC	Lleva a cabo una interpolación circular en 3D en el orden de punto de partida, punto de tránsito 1, punto de tránsito 2 y punto de inicio.
MVR	Lleva a cabo un movimiento de interpolación circular en 3D desde el punto inicial al punto final a través de los puntos de tránsito.

MVR2	Lleva a cabo un movimiento de interpolación circular en 3D desde el punto inicial al punto final pasando por los puntos de referencia.
MVR3	Lleva a cabo un movimiento de interpolación circular en 3D desde el punto inicial al punto final pasando un punto central.
MVS	Lleva a cabo el movimiento de interpolación lineal a partir de la posición actual hasta la posición final del movimiento.
OADL	Ajusta automáticamente la óptima aceleración/desaceleración en función del estado de carga de la mano de un robot.
OVRD	Especifica la velocidad del movimiento del robot del 1 al 100%. Esto se aplica a todo el programa.
SERVO	Controla encendido/apagado de los servos.
SPD	Designa la velocidad de los movimientos lineales y circulares del robot.
TOOL	Designa a la conversión de datos de la herramienta.
TORQ	Designa el límite de torque para cada eje.
WTH	Añade un proceso al movimiento de interpolación.
WTHIF	Añade un proceso condicionado al movimiento de interpolación.

Tabla 3.1.7.1.3 Instrucciones de movimiento

- **INSTRUCCIONES DE CONTROL DE PROGRAMA**

Estos comandos se refieren al control del comportamiento del programa a lo largo de la ejecución del código. Las instrucciones comunes se detallan en la siguiente tabla.

Comando	Función
ACT	Designa el activar / desactivar el estado de la alarma.
CALLP	Ejecuta el programa designado.
COM OFF	Designa desactivar las interrupciones de una línea de comunicación.
COM ON	Designa activar las interrupciones de una línea de comunicación.

COM STOP	Designa detener las interrupciones de una línea de comunicación.
DEF ACT	Esta instrucción establece las condiciones de interrupción de las señales de control concurrentes y lleva a cabo el procesamiento de interrupciones durante la ejecución del programa, así como el proceso que se llevará a cabo cuando se produce una interrupción.
END	Finaliza la ejecución de un programa.
FOR NEXT	Ejecuta repetidamente el programa entre el FOR y NEXT hasta que las condiciones finales sean cumplidas.
FPRM	Define el orden de los argumentos, el tipo, y el número para el programa principal que utiliza argumentos en un subprograma.
GOSUB	Llama a la subrutina en el N <sup>o</sup> de línea designado o etiqueta.
GOTO	Incondicionalmente salta al N <sup>o</sup> de línea designado o etiqueta.
HLT	Interrumpe la ejecución del programa y el movimiento del robot.
IF THEN ELSE	Un proceso es seleccionado y ejecutado de acuerdo a los resultados de una expresión.
ON COM GOSUB	Define el salto a una subrutina cuando se genera una interrupción de una línea de comunicación.
ON GOSUB	Llama a la subrutina en el número de línea o la etiqueta que corresponde al valor.
ON GOTO	Salta a la línea o la etiqueta que corresponde al valor.
PRIORITY	En la operación de un programa multitarea, múltiples líneas son ejecutadas en secuencia. Esta instrucción define la prioridad de ejecución.
RESET ERR	Resetea un error generado en la controladora del robot.
RETURN	Cuando se regresa de una subrutina normal vuelve a la línea siguiente a la línea del GOSUB. Cuando regresa de una subrutina de procesamiento de interrupciones, regresa a la línea donde se generó la interrupción, o en la línea siguiente.
SKIP	Transfiere el control del programa a la línea siguiente.
WAIT	Espera a que la variable alcance el valor deseado.

WHILE WEND	El programa entra en la instrucción WHILE y WEND y se repite hasta que las condiciones de bucle sean cumplidas.
XCLR	Esta instrucción cancela la tarea especificada dentro de un programa. Se utiliza durante el funcionamiento multitarea.
XLOAD	Esta instrucción carga la tarea especificada dentro de un programa.
XRST	La línea de ejecución del programa regresa a la línea superior.
XRUN	Ejecuta el programa designado en paralelo.
XSTP	Detiene el funcionamiento del programa en la ranura seleccionada.

Tabla 3.1.7.1.4 Instrucciones de movimiento

- **INSTRUCCIONES DE CONTROL DE MANO**

Comando	Función
HCLOSE	Cierra la mano del robot.
HOPEN	Abre la mano del robot.

- **INSTRUCCIONES DE ENTRADA/SALIDA**

Comando	Función
DEF IO	Declara una variable de entrada / salida.

## COMO CREAR UN NUEVO PROYECTO

En este capítulo se revisará como crear un nuevo proyecto en COSIROP. Una vez abierto el programa nos aparece una pantalla en gris, sin ningún control en particular.



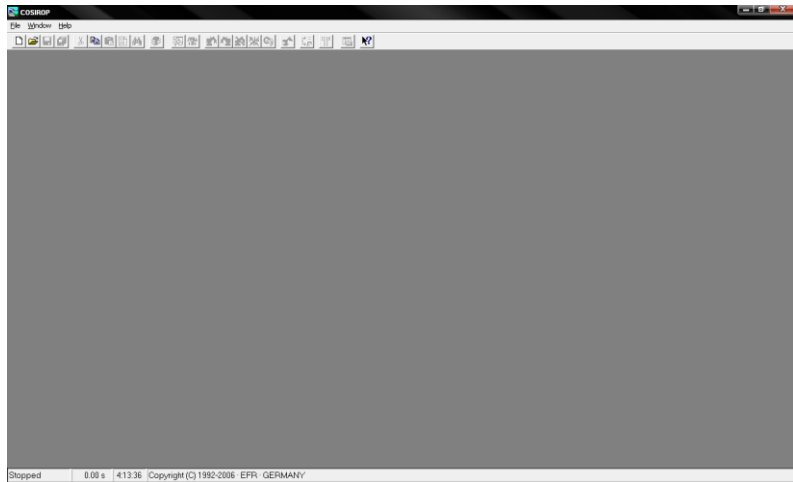


Figura 3.1.7.1.37 ventana de inicio de Cosirop<sup>74</sup>

Para abrir un nuevo proyecto vamos a File / Project Wizard

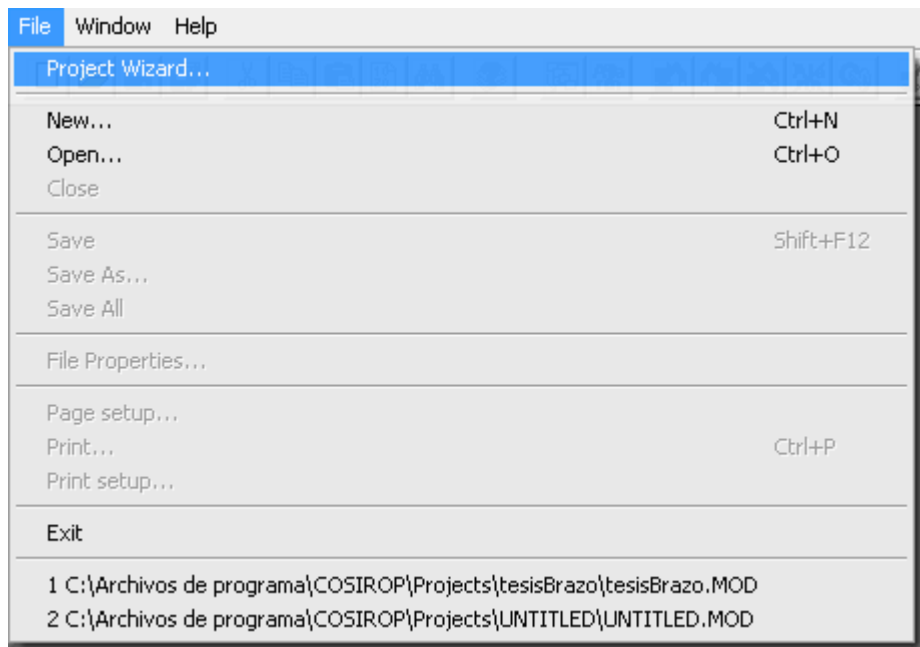


Figura 3.1.7.1.38 menú file de COSIROP<sup>75</sup>

Por defecto el nombre del proyecto aparece UNTITLED, ahí se pone el nombre que se desea para el proyecto. Se define el número de programa, tomando en cuenta

<sup>74</sup> COSIROP

<sup>75</sup> COSIROP

que ese es el número que toma en cuenta la controladora del robot. Se puede escoger el directorio en donde se va a guardar el proyecto y poner información adicional referente al proyecto.

Figura 3.1.7.1.39 paso 1 de Project Wizard<sup>76</sup>

Luego se escoge el tipo de robot con el cual se va a trabajar. Esto es muy importante porque según eso el programa hará la revisión de la sintaxis, ya que, a pesar de que todos utilicen el mismo lenguaje la forma de escribir las instrucciones varia de robot en robot. En esta ventana podemos definir que lenguaje va a utilizar, que cantidad de entradas/salidas tiene y si el robot tiene ejes adicionales a los convencionales de su tipo.

<sup>76</sup> COSIROP

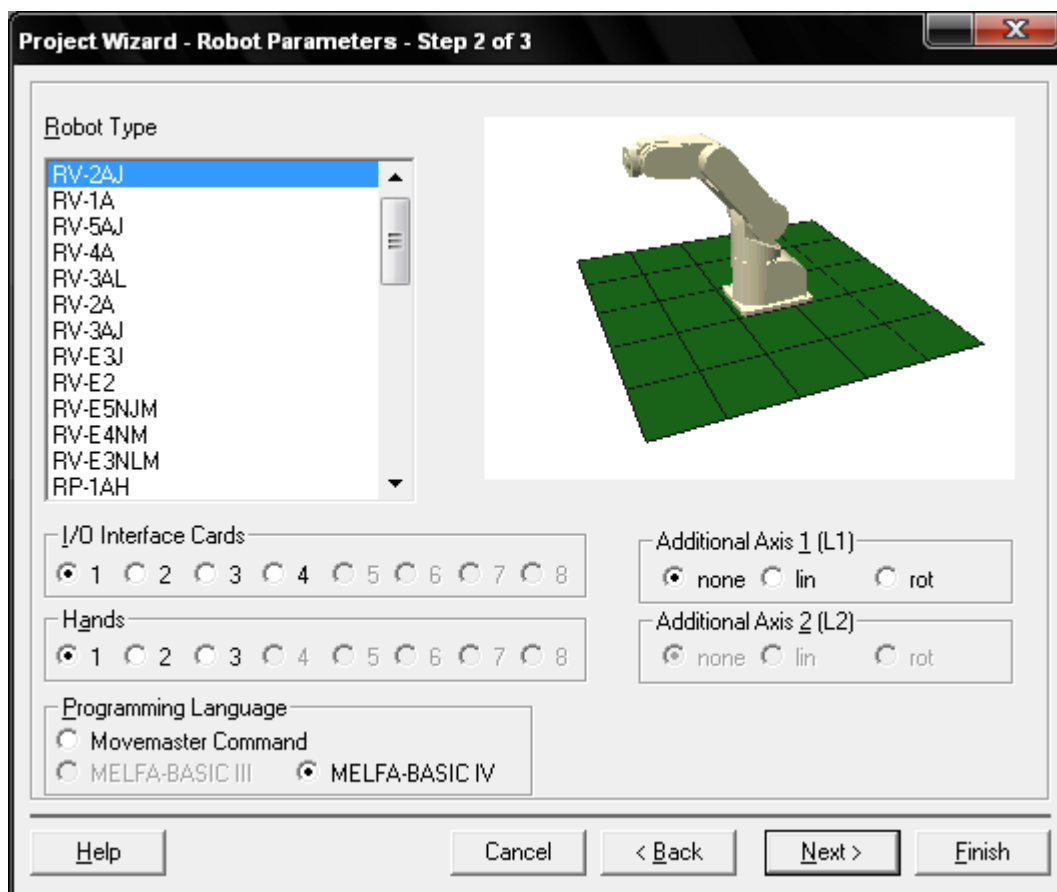


Figura 3.1.7.1.40 paso 2 de Project Wizard<sup>77</sup>

En este punto podemos presionar FINISH para terminar el wizard o se puede poner siguiente para ir al control de cambios en donde se registran los cambios que hemos realizado en las versiones del mismo programa. Si cuando presionemos FINISH aparece un cuadro como el siguiente

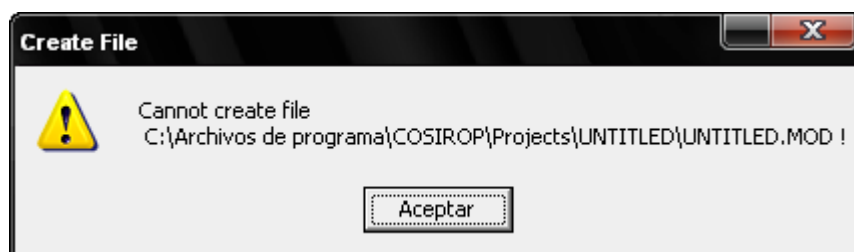


Figura 3.1.7.1.41 mensaje de alarma por proyecto existente<sup>78</sup>

<sup>77</sup> COSIROP

<sup>79</sup> COSIROP

Significa que el nombre que estamos utilizando ya existe y tenemos que cambiarlo, para ello regresamos a la pantalla inicial.

Una vez finalizado ya podemos empezar con el desarrollo de nuestro proyecto. Aparecerá una pantalla en blanco con diferentes ventanas de trabajo.

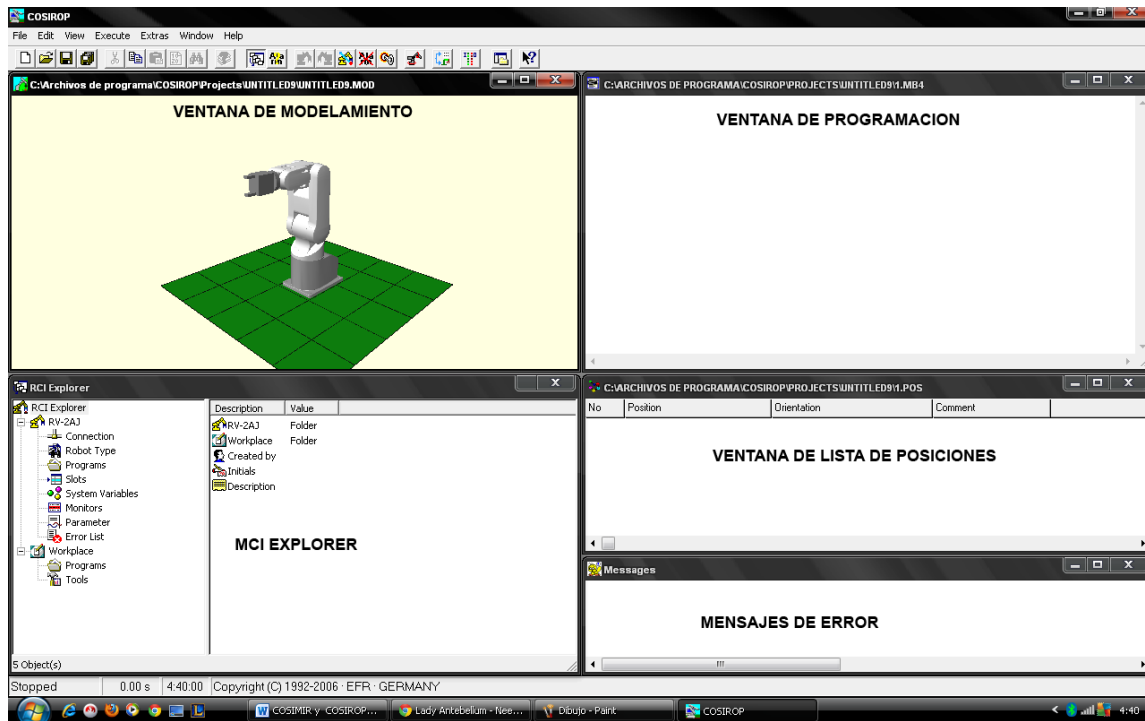


Figura 3.1.7.1.42 ventanas principales para programación<sup>79</sup>

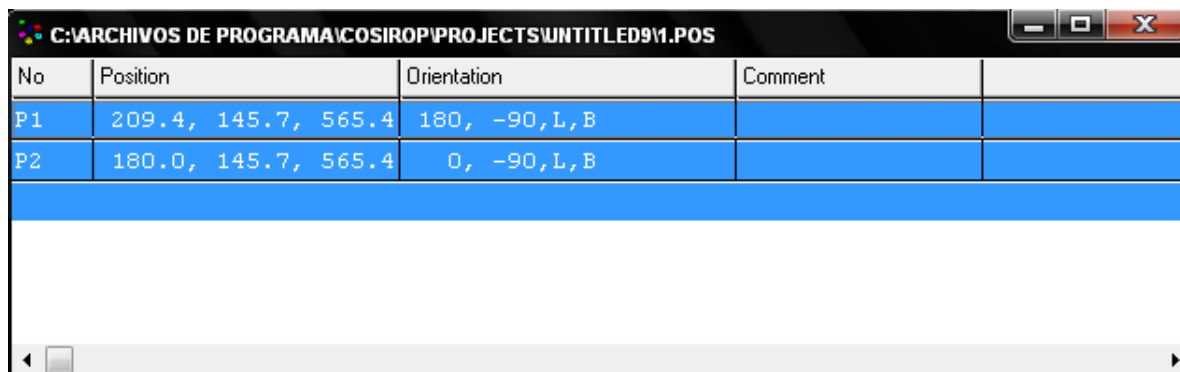
Cada una de estas ventanas ya han sido explicadas anteriormente, por lo que no se volverá a repetir, si se necesita más información revisar el inicio de este capítulo.

## EJEMPLO DE UN PROGRAMA EN COSIROP

En este capítulo se mostrará un programa simple para descargar al robot.

Antes de escribir cualquier línea de código hay que generar la lista de posiciones a las que hará referencia el programa.

<sup>79</sup> COSIROP



No	Position	Orientation	Comment	
P1	209.4, 145.7, 565.4	180, -90,L,B		
P2	180.0, 145.7, 565.4	0, -90,L,B		

Figura 3.1.7.1.43 ventana de lista de posiciones<sup>80</sup>

Luego se escribe un programa en la ventana de programación



```

MOV P1
DLY 100
MOV P2
DLY 100
CLOSEH 1
END

```

Figura 3.1.7.1.44 ventana de programa sin numerar<sup>81</sup>

Luego de eso se numera el código presionando el botón .

<sup>80</sup> COSIROP

<sup>81</sup> COSIROP



```

10 MOV P1
20 DLY 100
30 MOV P2
40 DLY 100
50 CLOSEH 1
60 END

```

Figura 3.1.7.1.45 ventana de programa numerado<sup>82</sup>

Una vez numerado el programa hay que compilarlo, presionando el botón Syntax

Check .

Hecho esto el programa no debe mostrar ningún error.

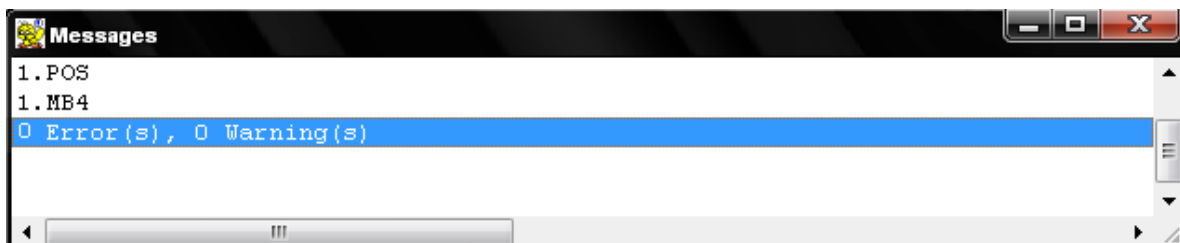


Figura 3.1.7.1.46 ventana de mensajes de error<sup>83</sup>

Ya que se tiene la lista de posiciones y el programa compilado sin error se procede a descargarlo a la controladora.

## COMO ESTABLECER UNA CONEXIÓN DEL PC A LA CONTROLADORA

Aquí se mostrarán los pasos necesarios para establecer una conexión exitosa desde el PC hacia la controladora del robot.

<sup>82</sup> COSIROP

<sup>83</sup> COSIROP

Lo primero que necesitamos hacer es revisar que todo esté conectado correctamente y encendido, es importante hacerlo en ese orden y no primero encender y luego revisar las conexiones.

Como primer paso debemos revisar que el cable de comunicación serial esté conectado a la controladora



Figura 3.1.7.1.47 conexión del cable serial a la controladora<sup>84</sup>

Una vez hecho esto encendemos la controladora desde su botón de encendido, presionándolo sobre su parte derecha.

---

<sup>84</sup> FUENTE LOS AUTORES



Figura 3.1.7.1.48 encendido de la controladora<sup>85</sup>

Luego verificamos que la llave de la controladora esté en modo “Auto Ext.” Y la llave del teach se encuentre en “Disable”.



Figura 3.1.7.1.49 colocación de la controladora en Auto Ext<sup>86</sup>

Ahora tenemos el sistema listo para ser controlado vía serial desde la computadora.

<sup>85</sup> FUENTE LOS AUTORES

<sup>86</sup> FUENTE LOS AUTORES



El siguiente paso es configurar los parámetros de comunicación serial.  
Para ello vamos al RCI EXPLORER y seleccionamos “Connection”.

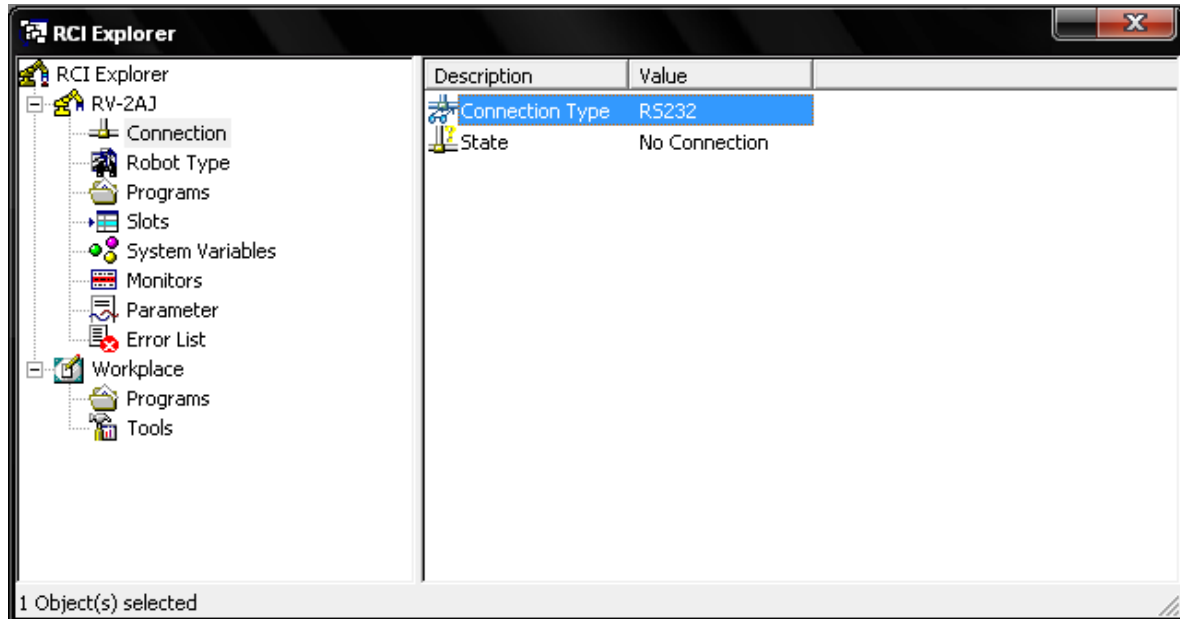


Figura 3.1.7.1.50 RCI Explorer<sup>87</sup>

En la ventana de la derecha seleccionamos Connection Type o State y le damos doble click, para que aparezca la ventana de configuración de parámetros. Seleccionamos la segunda pestaña que dice Serial Interface, ya que la comunicación que vamos a realizar es RS-232.

<sup>87</sup> COSIROP

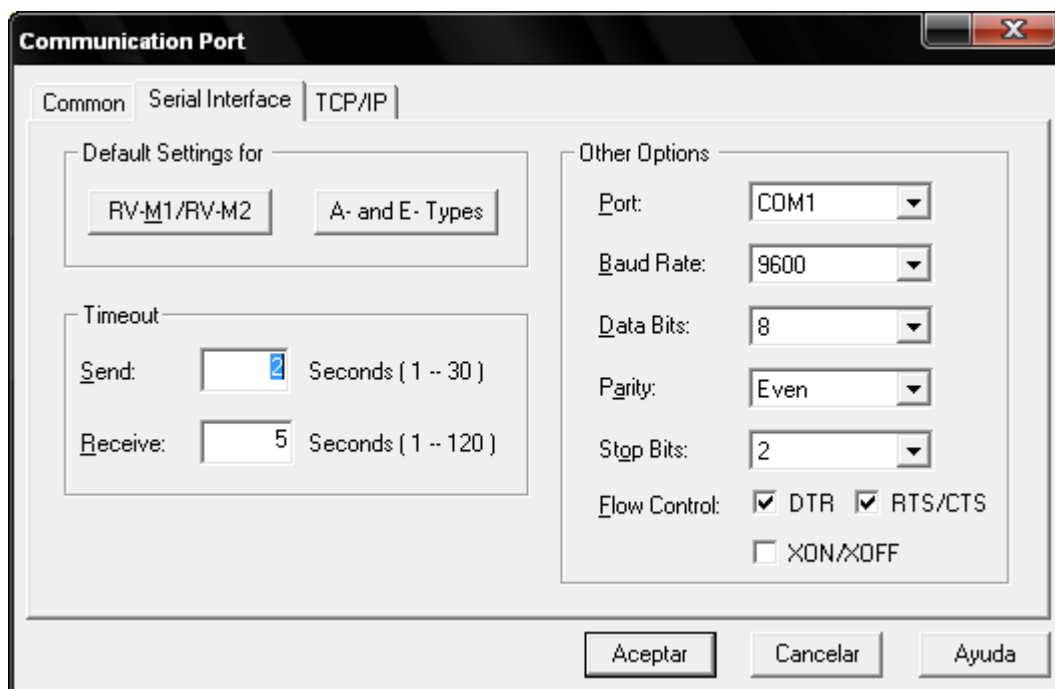


Figura 3.1.7.1.51 ventana de configuración de parámetros para comunicación serial<sup>88</sup>

Lo primero que hay que hacer es seleccionar el tipo de robot que vamos a utilizar, para esto tenemos 2 botones en la parte izquierda, se presiona el que dice “A – and E - Types”, ya que el robot es el RV-2AJ. Esto setea todos los parámetros de comunicación, excepto el puerto COM que hay que seleccionarlo manualmente.

Si se utiliza un PC de sobremesa el puerto será el COM1, pero si se utiliza una laptop con un driver USB-SERIAL, es necesario revisar que COM fue creado para la comunicación. Para ello hay que dar click derecho sobre MI PC y seleccionar propiedades. Este ejemplo está diseñado para Windows XP aunque funciona de manera similar para Windows Vista y Windows 7.

<sup>88</sup> COSIROP

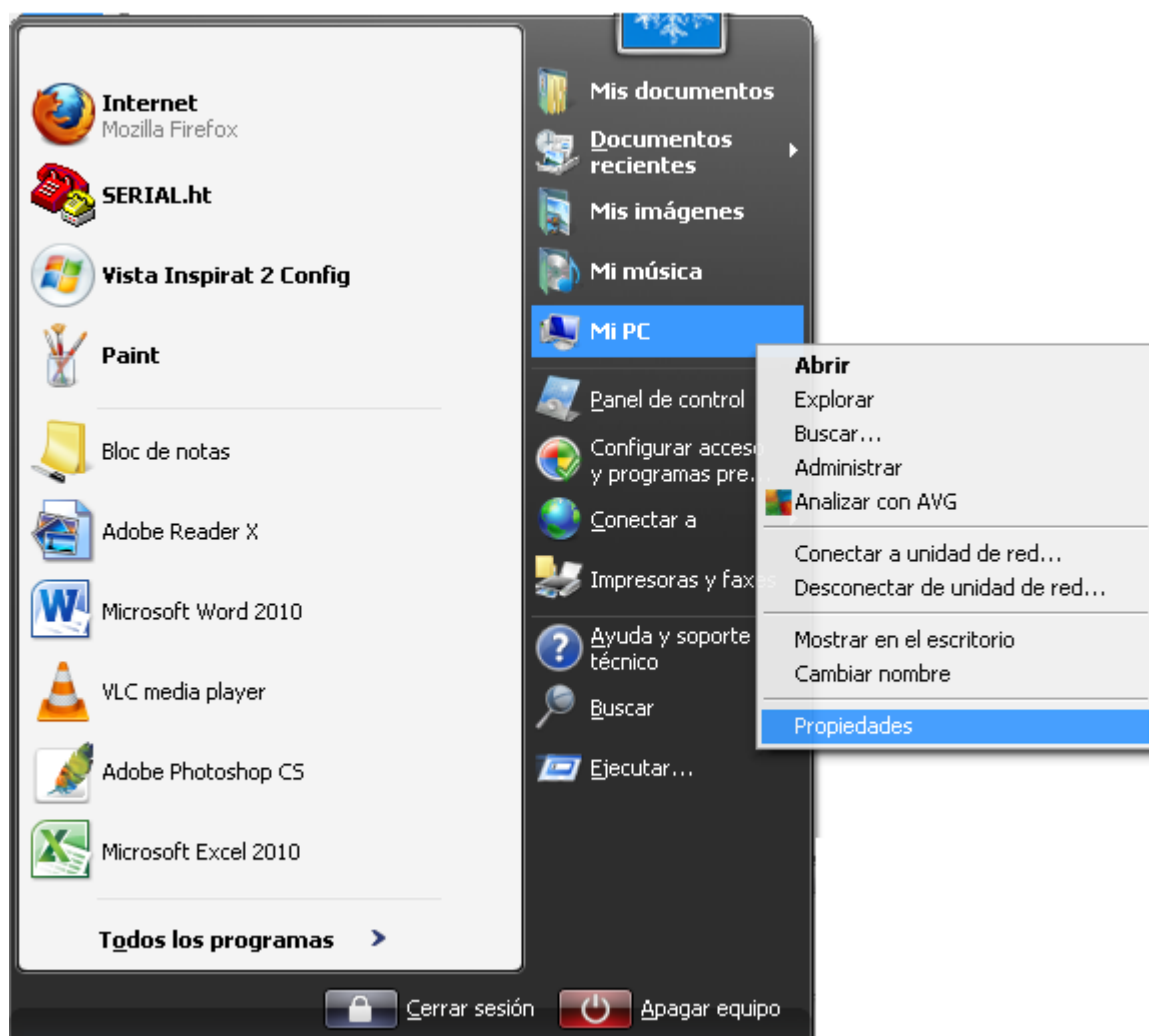


Figura 3.1.7.1.52 selección de propiedades de mi pc<sup>89</sup>

Esto despliega una ventana que da algunas opciones, de las cuales hay que seleccionar “HARDWARE”.

<sup>89</sup> FUENTE LOS AUTORES

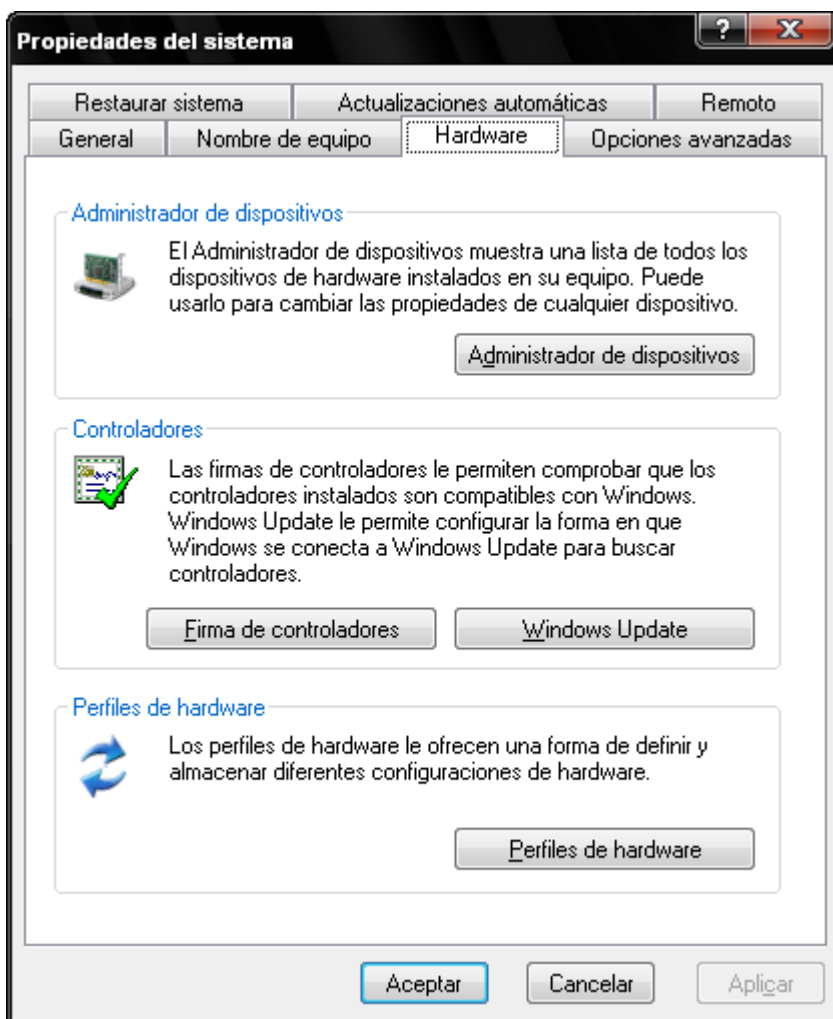


Figura 3.1.7.1.53 propiedades de sistema Windows XP<sup>90</sup>

En esta pestaña hay que presionar el botón “Administrador de dispositivos”. Que muestra información de los diferentes periféricos existentes en la laptop. Hay que revisar los puertos COM e identificar el correspondiente al driver que se esté utilizando.

<sup>90</sup> FUENTE LOS AUTORES

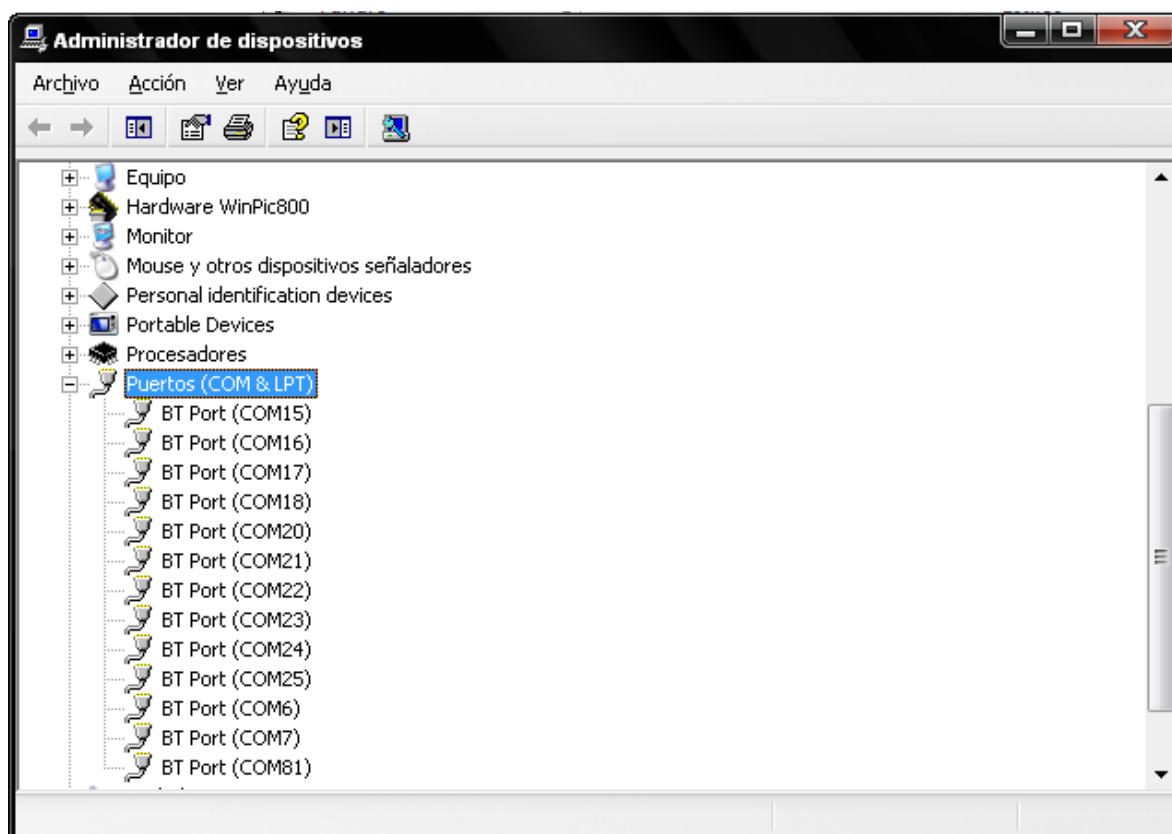


Figura 3.1.7.1.54 administrador de dispositivos Windows XP<sup>91</sup>

Una vez identificado el COM que corresponde al driver USB-SERIAL, se selecciona ese mismo puerto en la ventana de configuración en el combo que dice “Port”.

---

<sup>91</sup> FUENTE LOS AUTORES

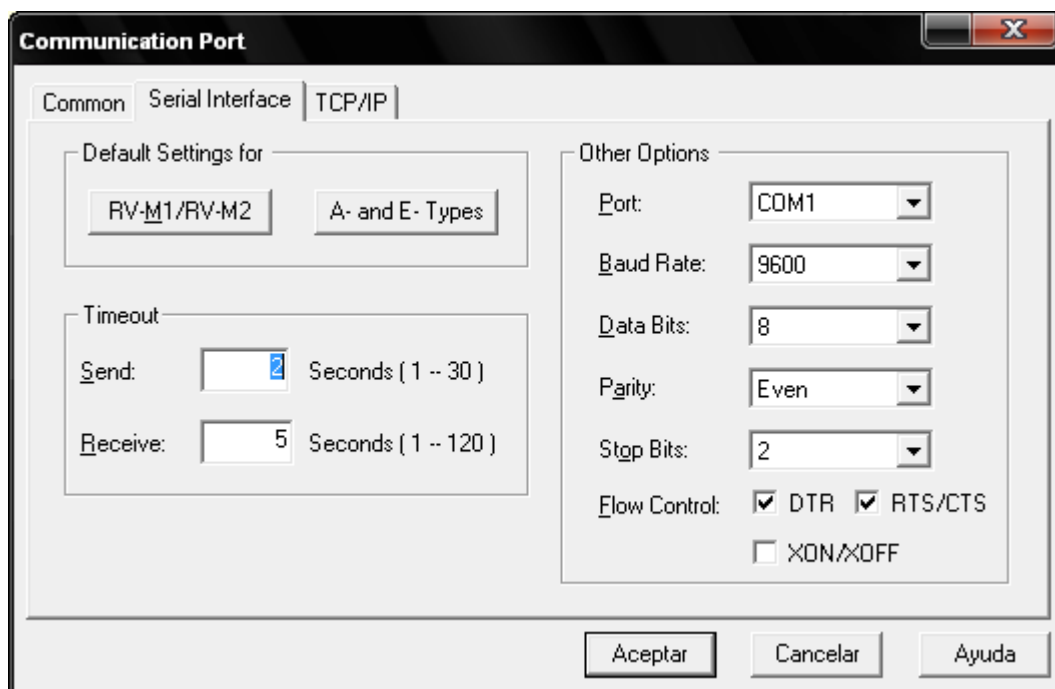


Figura 3.1.7.1.55 ventana de configuración de parámetros para comunicación serial<sup>92</sup>

## COMO PROBAR QUE LA COMUNICACIÓN FUE EXITOSA

Para comprobar que todo está correcto basta con ir al RCI EXPLORER y dar click derecho sobre “Connection” y seleccionar “Connect”.

<sup>92</sup> COSIROP

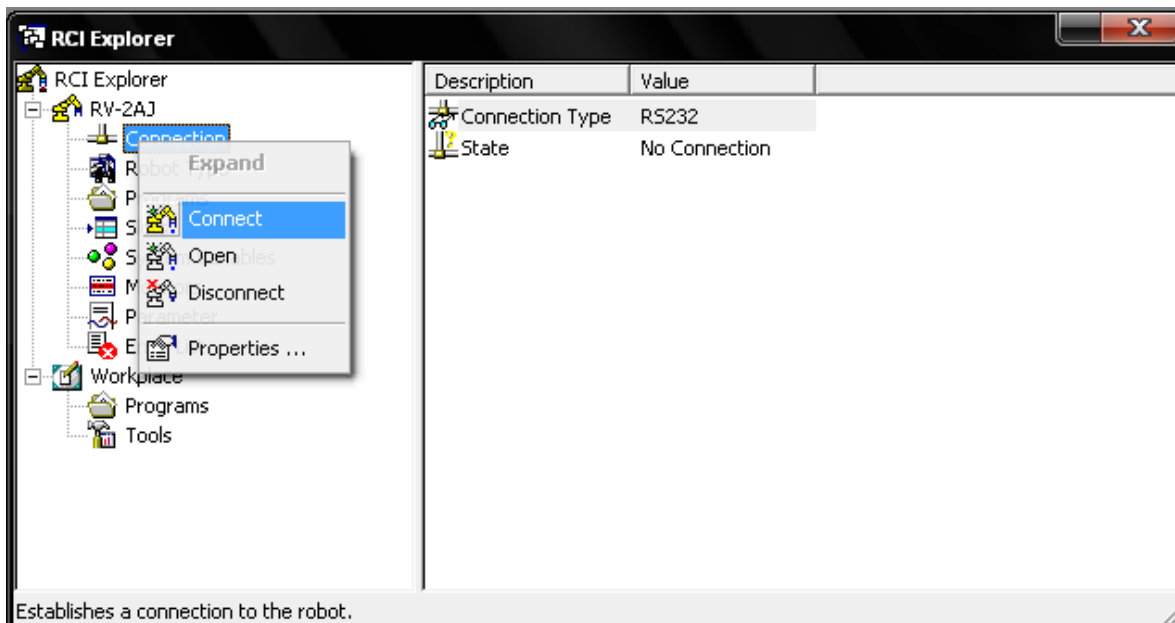



Figura 3.1.7.1.56 RCI EXPLORER <sup>93</sup>

Si todo está correcto el estado cambiará de “No Connection” a “Conected”. Si esto no ocurre hay que revisar que todas las conexiones físicas estén correctas y volver a revisar los pasos de conexión antes mencionados.

### COMO DESCARGAR UN PROGRAMA Y COMO LEER UN PROGRAMA

Una vez lograda una conexión exitosa con la controladora, procedemos a descargar el programa al robot.

El proceso de descarga se lleva a cabo en 2 pasos. El primero es la descarga del programa. Para esto hay que seleccionar la ventana del programa en donde se

encuentra el código y presionar el botón  Download PC->Robot, que muestra la siguiente ventana de descarga.

<sup>93</sup> COSIROP

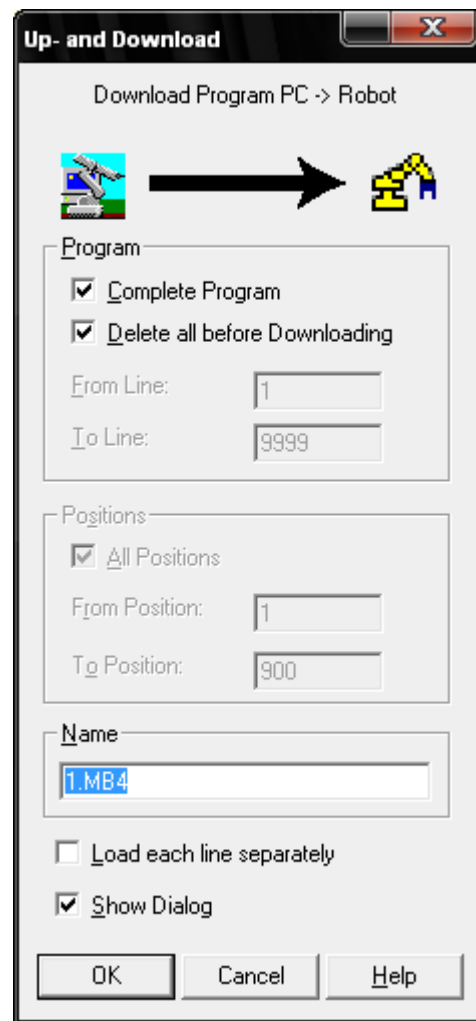



Figura 3.1.7.1.57 ventana de descarga de programa del PC al robot<sup>94</sup>

En la parte marcada con azul, se muestra el nombre con el que el programa se va a descargar a la controladora. Hay que tener cuidado de no sobrescribir ningún programa existente o importante para el robot. Si no se está seguro es preferible cambiar el nombre del programa. Solamente se permiten nombres numéricos para los programas, ya que la controladora no entiende otro tipo de nomenclatura.

Una vez descargado el programa hay que descargar al lista de posiciones, para esto se selecciona la ventana de posiciones y se vuelve a presionar el botón 

<sup>94</sup> COSIROP



Download PC->Robot. Esto vuelve a mostrar la ventana de descarga, pero con una extensión diferente.

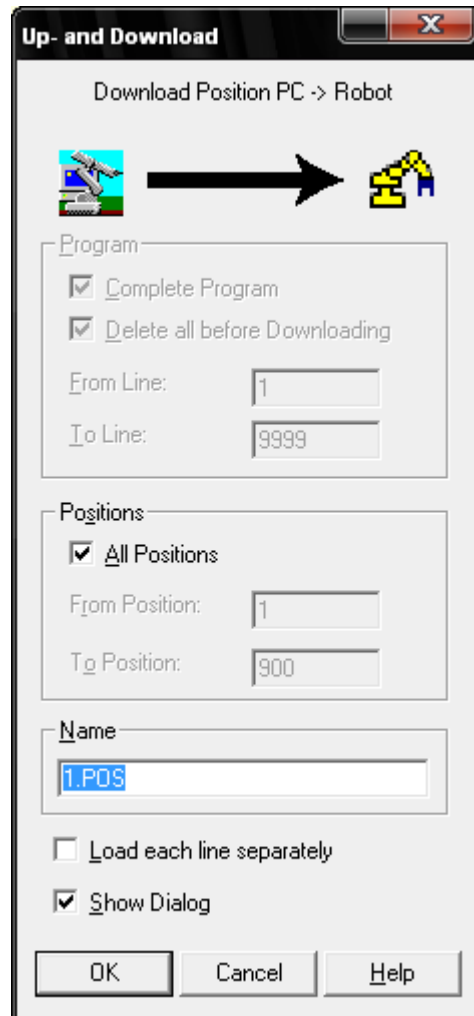


Figura 3.1.7.1.58 ventana de descarga de lista de posiciones del PC al robot<sup>95</sup>

Hay que tener presente que para que el programa funcione correctamente en el robot. El archivo de la lista de posiciones debe tener el mismo nombre que el archivo de programa, porque si no al momento de ejecutar una alarma se disparará en la controladora.

<sup>95</sup> COSIROP

### **3.2 DESARROLLO DEL SOFTWARE**

Para determinar las funcionalidades del sistema se considera como premisa la construcción de estructuras que determinen los pasos a seguir y los procedimientos que se deben tomar en cuenta para generar una correcta arquitectura del proyecto.

### 3.2.1 CASOS DE USO

Se realizará la diagramación de los casos de uso del sistema.

#### 3.2.1.1 CASO DE USO 1: PRESENTACION DE CLASES DENTRO DEL PROGRAMA

##### AGENTES: USUARIO, CONTROLADORA

Se presenta la configuración del sistema a nivel de protocolos donde se va a escoger parámetros de inicio para la controladora y el Brazo Robótico RV-2AJ, estos parámetros son almacenados en un archivo encriptado para evitar cambios de configuración innecesarios.

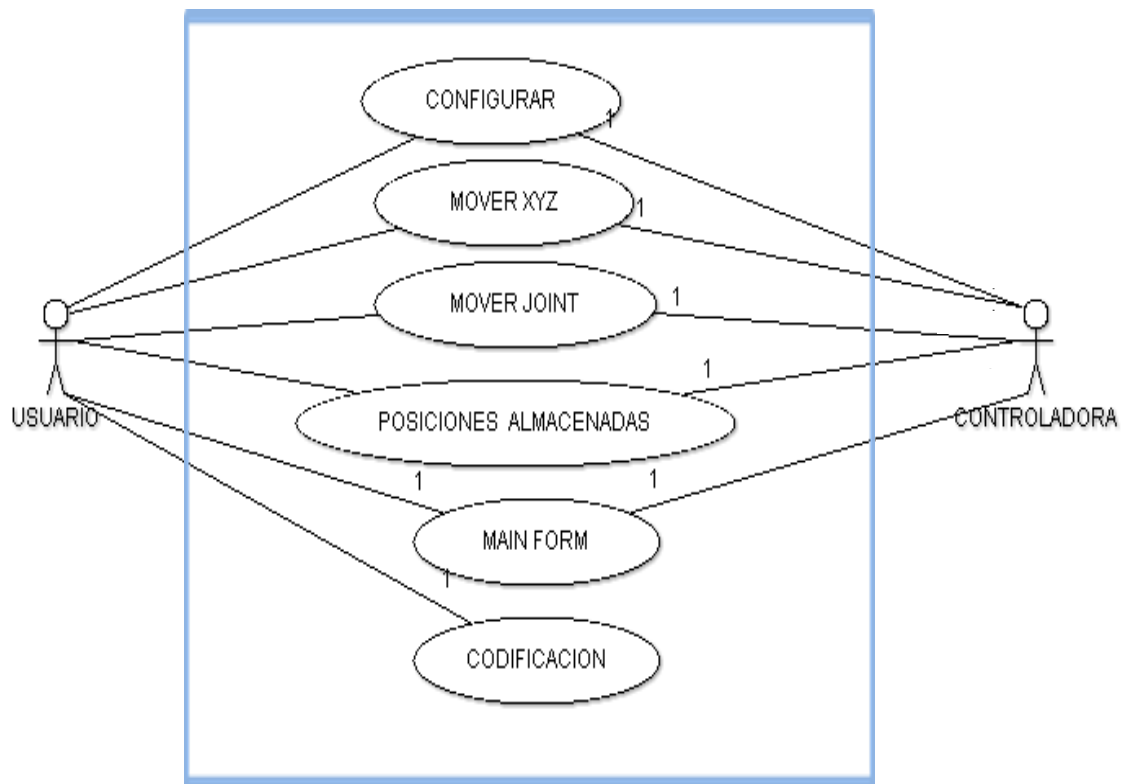


Figura 3.2.1.1.1 Caso de Uso 1: “Presentación de clases del programa” <sup>96</sup>

<sup>96</sup> Fuente Los Autores

Dentro de las interfaces se configuran parámetros para los movimientos en XYZ así como también para movimientos en modo JOINT, que permiten generar desplazamientos al Robot.

Se puede recurrir a posiciones almacenadas para desplazarse entre puntos almacenados en la controladora del brazo, realizando una lectura previa de los programas que se encuentran guardados.

Se establece una función para manejar eventos, posiciones del brazo en los ejes y ángulos correspondientes, y también se definen parámetros de comunicación.

### 3.2.1.2 CASO DE USO 2: CONFIGURACIÓN DE PROTOCOLOS DE COMUNICACIÓN

**AGENTES: USUARIO, CONTROLADORA.**

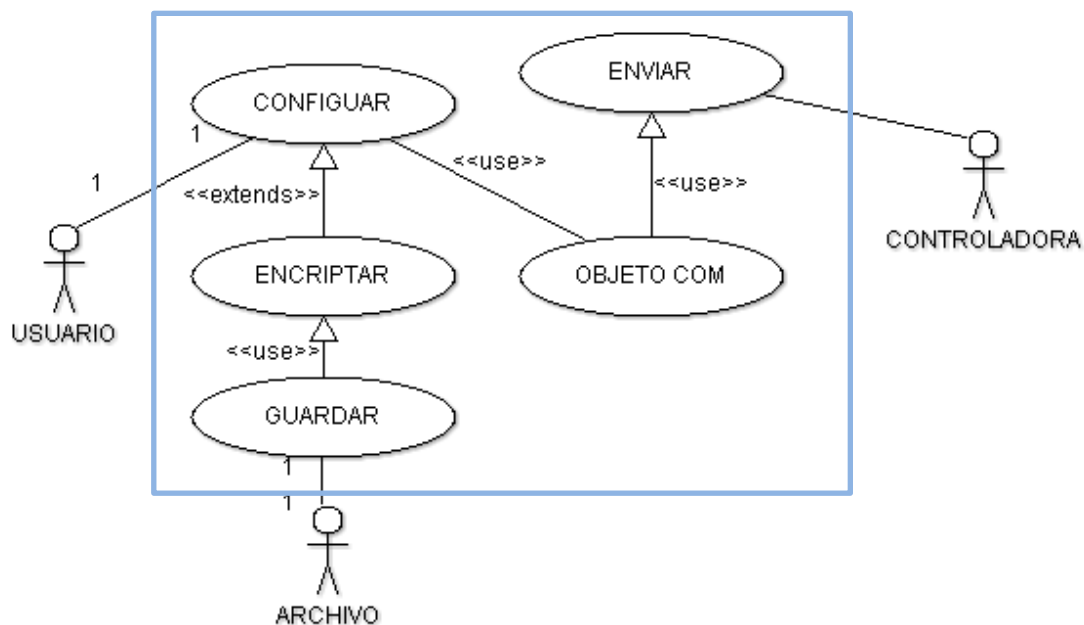


Figura 3.2.1.2.1 Caso de Uso 2: “Configuración de protocolos de comunicación”<sup>97</sup>

<sup>97</sup> Fuente los Autores

Se establece la comunicación con la controladora y se definen los parámetros de comunicación, puertos COM, paridad, Baudios, Bits de inicio, los cuales permiten tener la conexión con el Brazo y luego esa información se guarda en un texto encriptado.

### 3.2.1.3 CASO DE USO 3: MOVIMIENTOS EN XYZ AGENTES: USUARIO, CONTROLADORA.

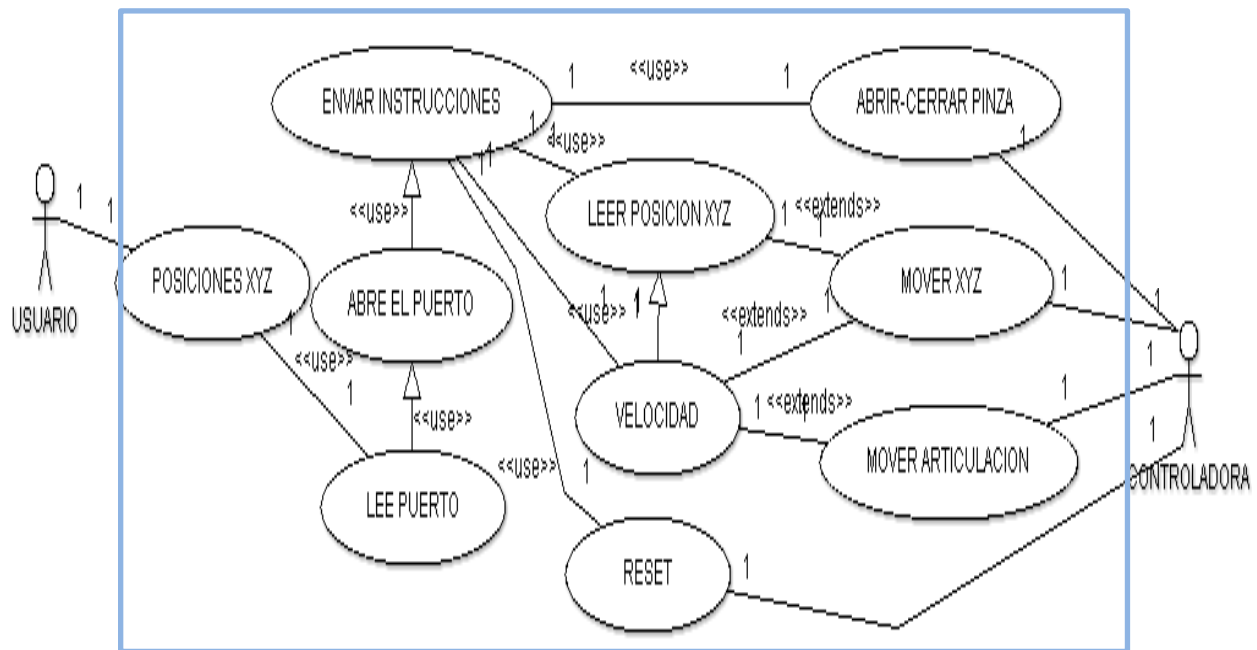


Figura 3.2.1.3.1 Caso de Uso 3: “MOVIMIENTOS EN XYZ”<sup>98</sup>

Para tener movimientos en el plano cartesiano se recurre a los desplazamientos en XYZ.

Se definen parámetros de velocidad y desplazamientos para tener un control del brazo sobre la estación de trabajo.

<sup>98</sup> Fuente los Autores

### 3.2.1.4 CASO DE USO 4: MOVIMIENTOS EN MODO JOINT

AGENTES: USUARIO, CONTROLADORA.

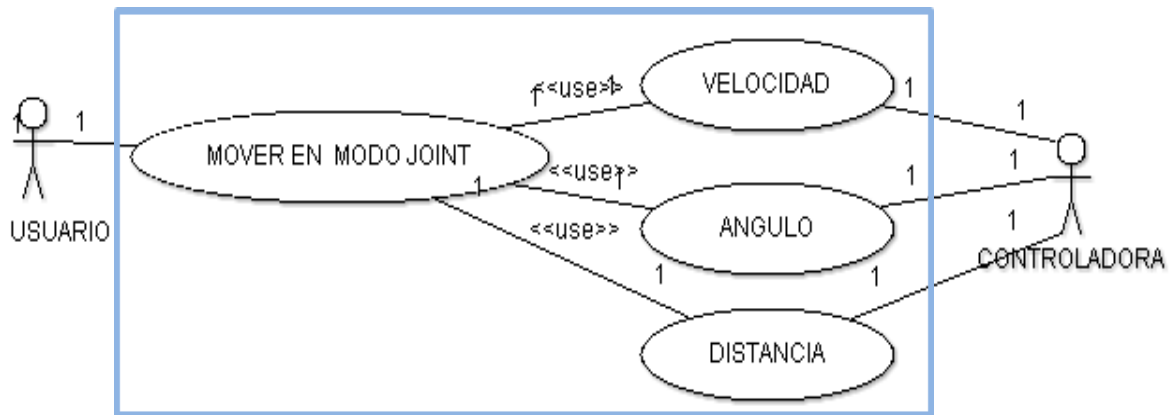


Figura 3.2.1.4.1 Caso de Uso 4: “MOVIMIENTOS EN JOINT”<sup>99</sup>

Para tener movimientos en función de sus articulaciones y ángulos se recurre a los desplazamientos en modo JOINT.

Se definen parámetros de velocidad y ángulos de desplazamientos para tener un control del brazo sobre la estación de trabajo.

<sup>99</sup> Fuente los Autores

**3.2.1.5 CASO DE USO 5: POSICIONES ALMACENADAS EN LA CONTROLADORA**  
**AGENTES: USUARIO, CONTROLADORA.**

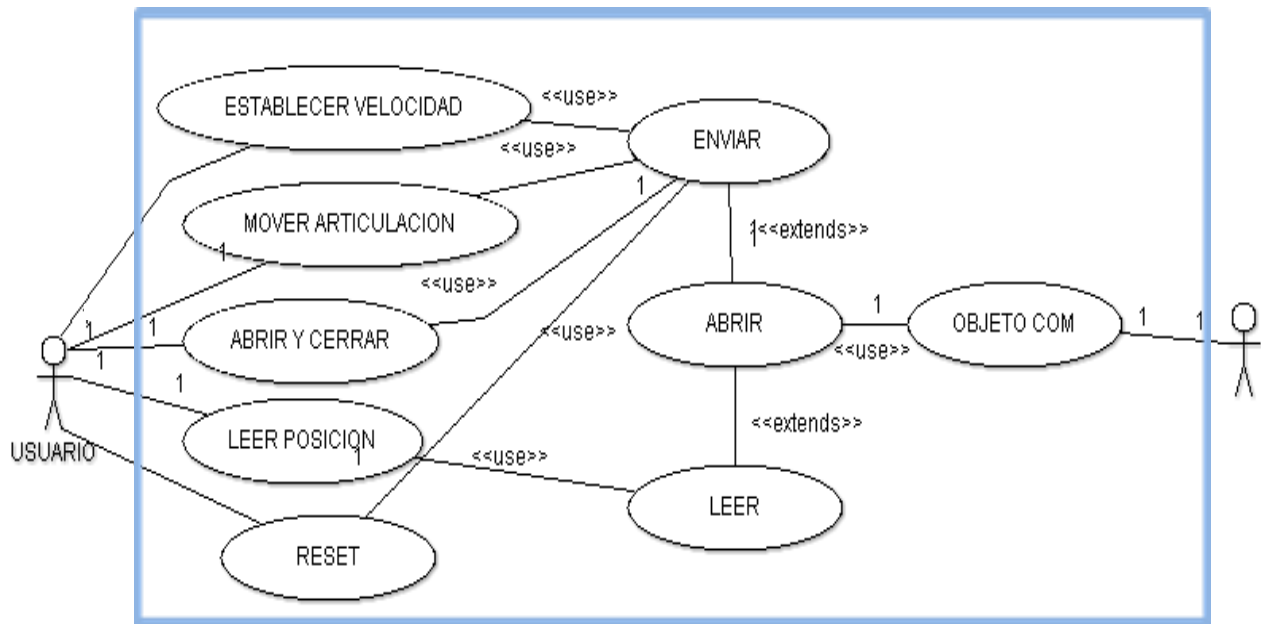


Figura 3.2.1.5.1 Caso de Uso 5: “MOVIMIENTOS ALMACENADOS EN LA CONTROLADORA”<sup>100</sup>

Las posiciones que están almacenadas en la controladora son cargadas en el sistema para listar los programas que se escogen de manera dinámica y se determina la velocidad para el movimiento de las articulaciones.

Las articulaciones se desplazan en las posiciones que son seleccionadas en el sistema.

Si existe alguna posición que el Brazo no pueda alcanzar se genera una alerta de error que se puede detener accionando el comando reset.

<sup>100</sup> Fuente los Autores

Se define el envío de instrucciones para controlar la velocidad y declarar las peticiones de mover en XYZ y mover articulaciones

Si se genera una alarma por alguna instrucción incorrecta se envía un comando de reset a la controladora. Se genera la instrucción para la habilitación de apertura y cierre del efector final.

Para realizar los comandos de envío y respuesta de la controladora se necesita establecer y definir los comandos de configuración sin modificar las variables que se declaran en el API.

### 3.2.1.6 CASO DE USO 6: CODIFICACIÓN

**AGENTES: USUARIO, CONTROLADORA.**

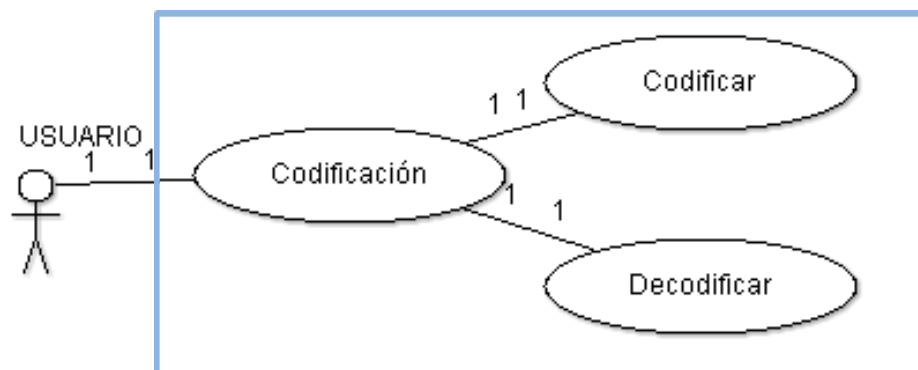


Figura 3.2.1.6.1 Caso de Uso 6: “CODIFICACIÓN” <sup>101</sup>

En la clase codificar se toma la información de las configuraciones de ingreso en el sistema con respecto a la comunicación, puerto COM, bit de parada, BaudRate, y se las guarda en un archivo plano codificado con Base64 para evitar cambios innecesarios en las configuraciones por default del sistema.

<sup>101</sup> Fuente los autores



La función codificar, toma los datos ingresados en el sistema y los transforma en Base64 y los guarda.

La función decodificar toma los datos del archivo plano y los descripta mediante el algoritmo, para ser utilizados por el sistema.

### 3.2.2 DIAGRAMAS DE CLASES

Los diagramas de clases nos permiten entender las diferentes relaciones y dependencias para manejar los eventos y llamados de las funciones principales del programa.

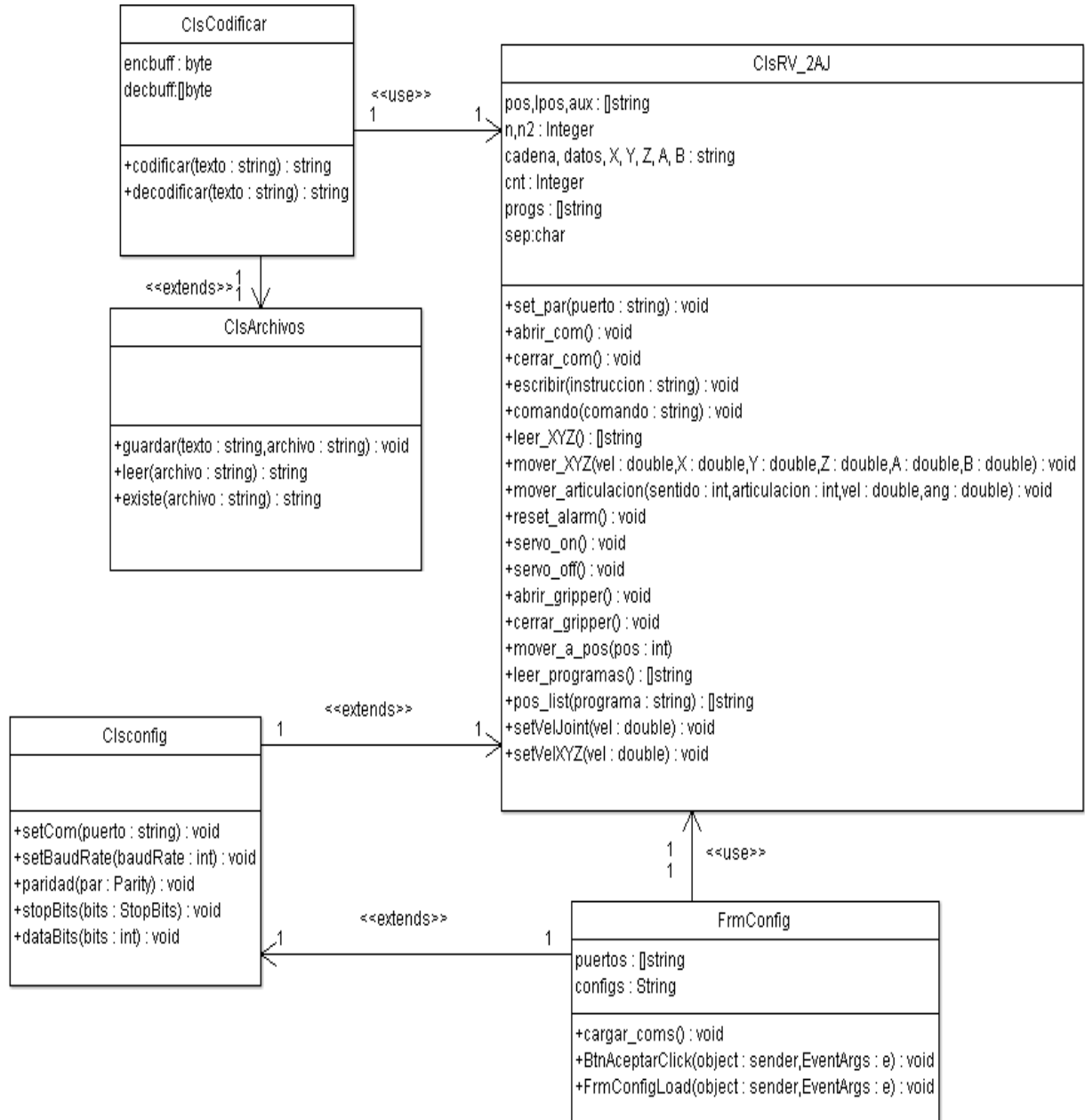


Figura 3.2.2.1 “DIAGRAMA DE CLASES”<sup>102</sup>

<sup>102</sup> Fuente los Autores

Se definen las clases principales para satisfacer las conexiones y comunicaciones del programa con la controladora del brazo y de ésta a la vez con el Brazo Robótico Mitsubishi **RV-2AJ**.

### 3.2.2.1 CLASE: ClsCodificar – CODIFICAR LA INFORMACIÓN

Clase para manejar la encriptación de los protocolos de comunicación con el sistema en Base64.

Atributos	
encbuff	Guarda el array codificado
decbuff	Guarda el array decodificado
Métodos	
Codificar()	Método para encriptar en Base64
Decodificar()	Método para desencriptar de Base64

### 3.2.2.2 CLASE: ClsArchivos - MANEJO DE ARCHIVOS

Clase para realizar el manejo de archivos donde se encuentran los parámetros en un texto plano.

Métodos	
guardar()	Método para guardar en un texto plano los parámetros de comunicación
leer()	Método para leer un archivo plano donde están los parámetros de comunicación.
existe()	Método para crear un archivo plano

### 3.2.2.3 CLASE: ClsConfig - CONFIGURACIONES

Clase para establecer los parámetros de la comunicación con la controladora

Métodos	
setCom()	Configura el puerto de comunicación
setBaud()	Configura el número de unidades de señal por segundo
paridad()	Configura los parámetros de transmisión
stopBits()	Configura los bit de parada
dataBits()	Configura los datos de envío

### 3.2.2.4 CLASE ClsRV\_2AJ - PRINCIPAL

Se establecen parámetros de comunicación, métodos de envío y petición de posiciones y programas con la controladora para mover al Brazo RV-2AJ

Atributos	
pos, lpos, aux	Variables tipo array para almacenar las posiciones
n,n2	Variable de finalización de línea
cadena, datos, X,Y,Z,A,B	Variables de control de cadena de caracteres para almacenar comandos de programación para la controladora
cnt	Variable de contador de posiciones
progs	Variable para almacenar en una

	cadena los programas
sep	Variable para almacenar los separadores de cadenas de respuesta de la controladora
<b>Métodos</b>	
set_par()	Determina los parámetros de configuración de la comunicación con la controladora
abrir_com()	Abre el puerto de comunicación con la controladora
cerrar_com()	Cierra el puerto de comunicación con la controladora
escribir()	Parámetros de envío default para comunicarse con la controladora
Comando()	Comandos de envío para establecer o cerrar las cadenas de I/O.
leer_XYZ()	Llamada para establecer la última posición del Brazo RV-2AJ en el plano de coordenadas XYZ
mover_articulacion()	Llamada para generar movimientos en las juntas del Brazo RV-2AJ(por ángulos)
mover_XYZ()	Llamada para generar movimientos del Brazo RV-2AJ en el plano de coordenadas XYZ
reset_alarm()	Llamada para reiniciar las funciones de la controladora.
mover_a_pos()	Método para desplazarse entre

	posiciones
leer_programas()	Método para establecer las coordenadas .
pos_list()	Recibe y enlista las posiciones del brazo RV-2AJ
setVelJoint()	Método para establecer la velocidad del brazo en JOINT
setVelXYZ()	Método para establecer la velocidad del brazo en XYZ

Tabla 3.3.2.1 Tabla de la clase principal

### 3.2.3 DIAGRAMAS DE SECUENCIA

Los diagramas de secuencia permiten distinguir las operaciones que se van ejecutando en las diferentes etapas del sistema, de esta manera se generan varios tipos de procesos donde se van a establecer los principales.

### 3.2.3.1 DIAGRAMA DE SECUENCIA 1: POSICIONES ALMACENADAS

Permite visualizar el control del proceso desde la lectura de la posición que se encuentra almacenada en la controladora hasta listar los programas y posiciones de las combinaciones de programas que se encuentran estructuradas y descargadas en la memoria del dispositivo de interconexión.

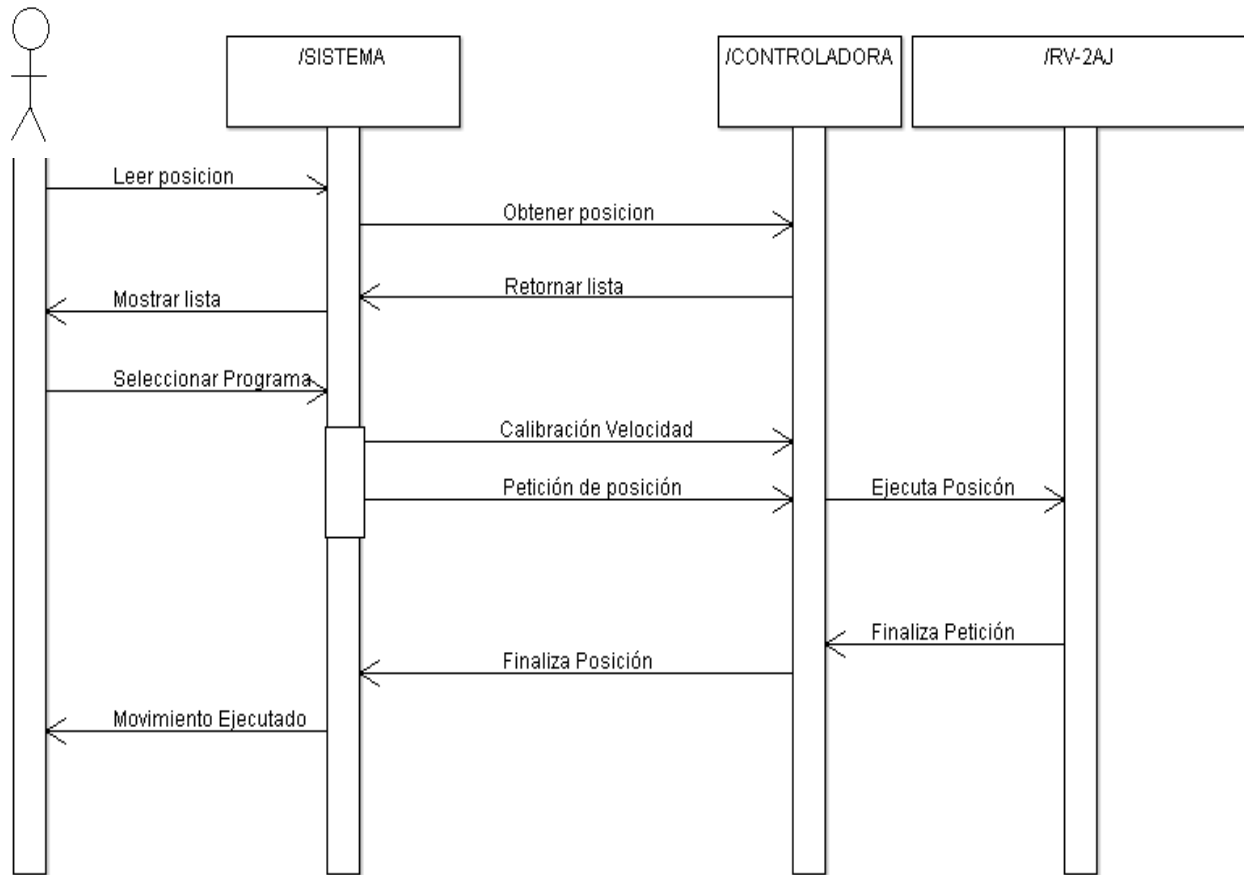


Figura 3.2.3.1.1 Diagrama de secuencia 1: “POSICIONES ALMACENADAS”<sup>103</sup>

<sup>104</sup> Fuente los Autores

### 3.2.3.2 DESCRIPCIÓN DE SECUENCIA 2: GENERACIÓN DE MOVIMIENTO JOINT

Permite visualizar el control que se establece en modo JOINT, donde se generan desplazamientos angulares, ya que se necesita mover articulaciones en ejes polares, se establece el rango en Grados, y la velocidad de manera porcentual

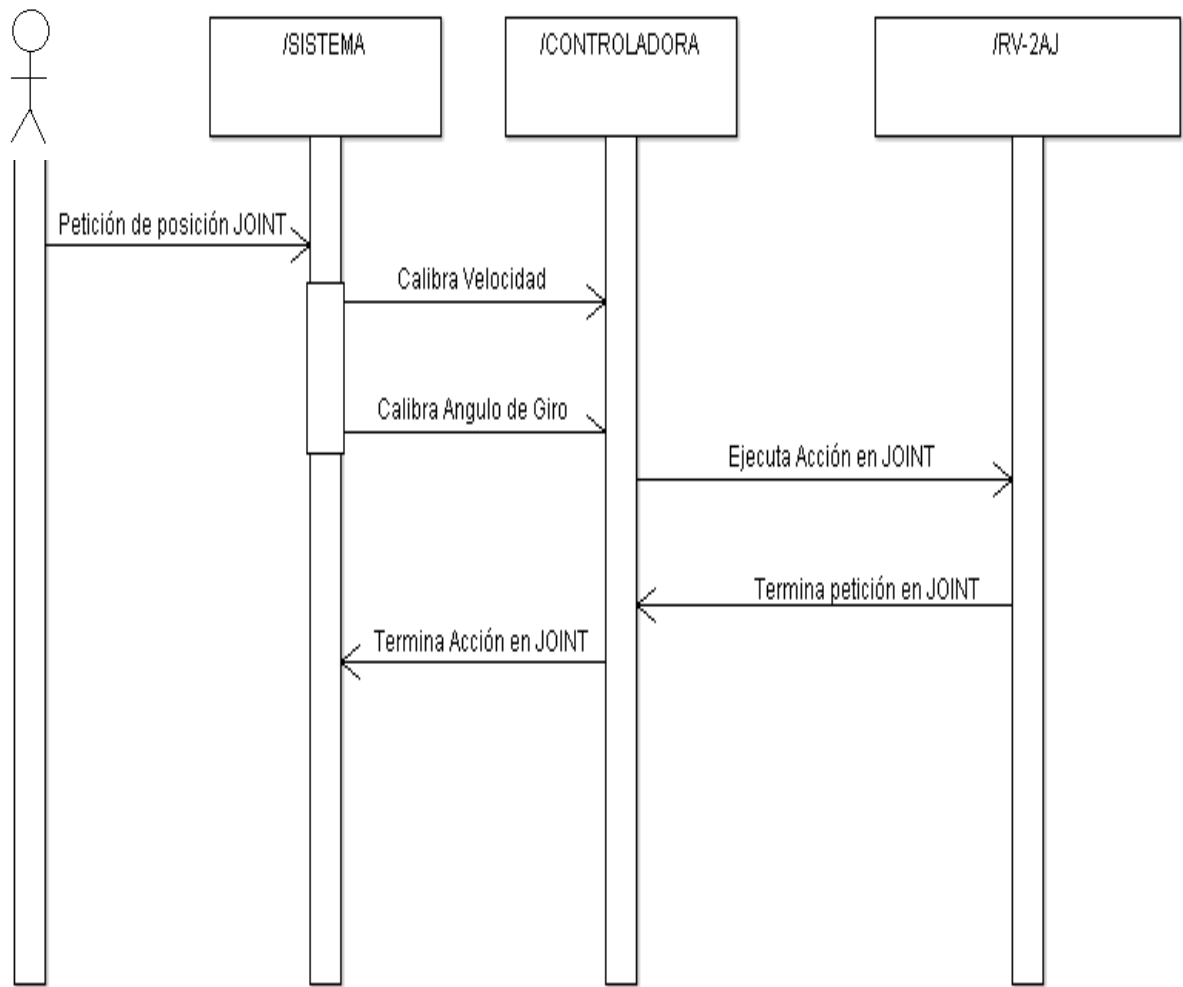


Figura 3.2.3.2.1 Diagrama de secuencia 2: “GENERACIÓN DE MOVIMIENTO JOINT” <sup>104</sup>

<sup>104</sup> Fuente los Autores



### 3.2.3.3 DESCRIPCIÓN SECUENCIA 3: GENERACIÓN DE MOVIMIENTO XYZ

Permite visualizar el control que se establece en modo cartesiano XYZ, donde se generan desplazamientos planos, ya que se necesita mover articulaciones en ejes coordenados, se establece el rango de desplazamiento en mm, y la velocidad de manera porcentual en mm/s.

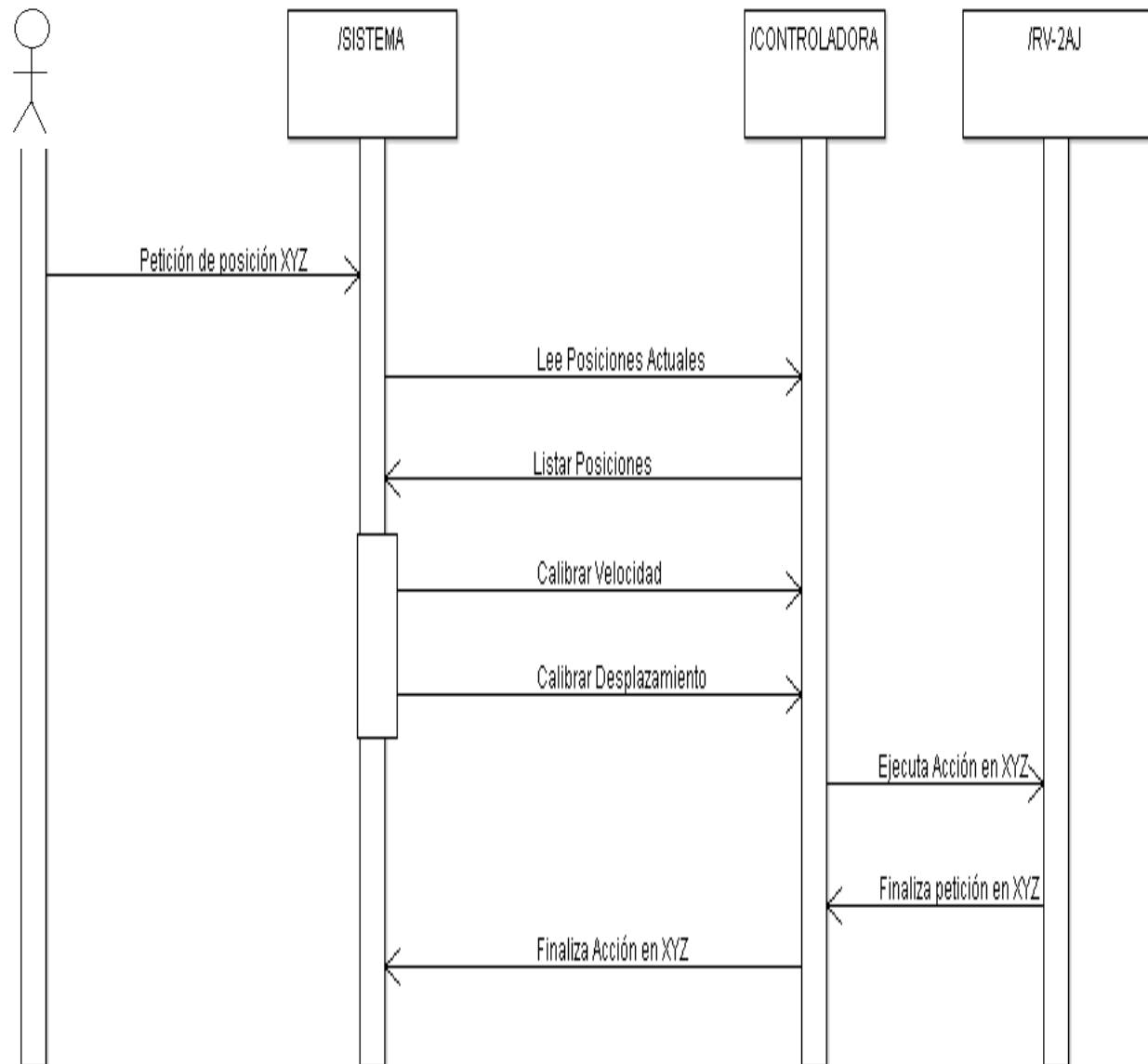


Figura 3.2.3.3.1 Diagrama de secuencia 2: “GENERACIÓN DE MOVIMIENTO JOINT”<sup>105</sup>

<sup>105</sup> Fuente los Autores

### 3.3 MODELADO DE INTERFAZ

Aquí se muestran las partes que tendrá la interfaz al final del proyecto.

#### 3.3.1 PANTALLA PRINCIPAL

Esta pantalla permitirá a los usuarios escoger a través de las siguientes opciones, las cuales ayudaran a controlar y configurar el Brazo RV-2AJ.

Dentro de cada botón se desplegarán opciones para poder interactuar con el brazo y realizar desplazamientos en diferentes tipos de movimientos.

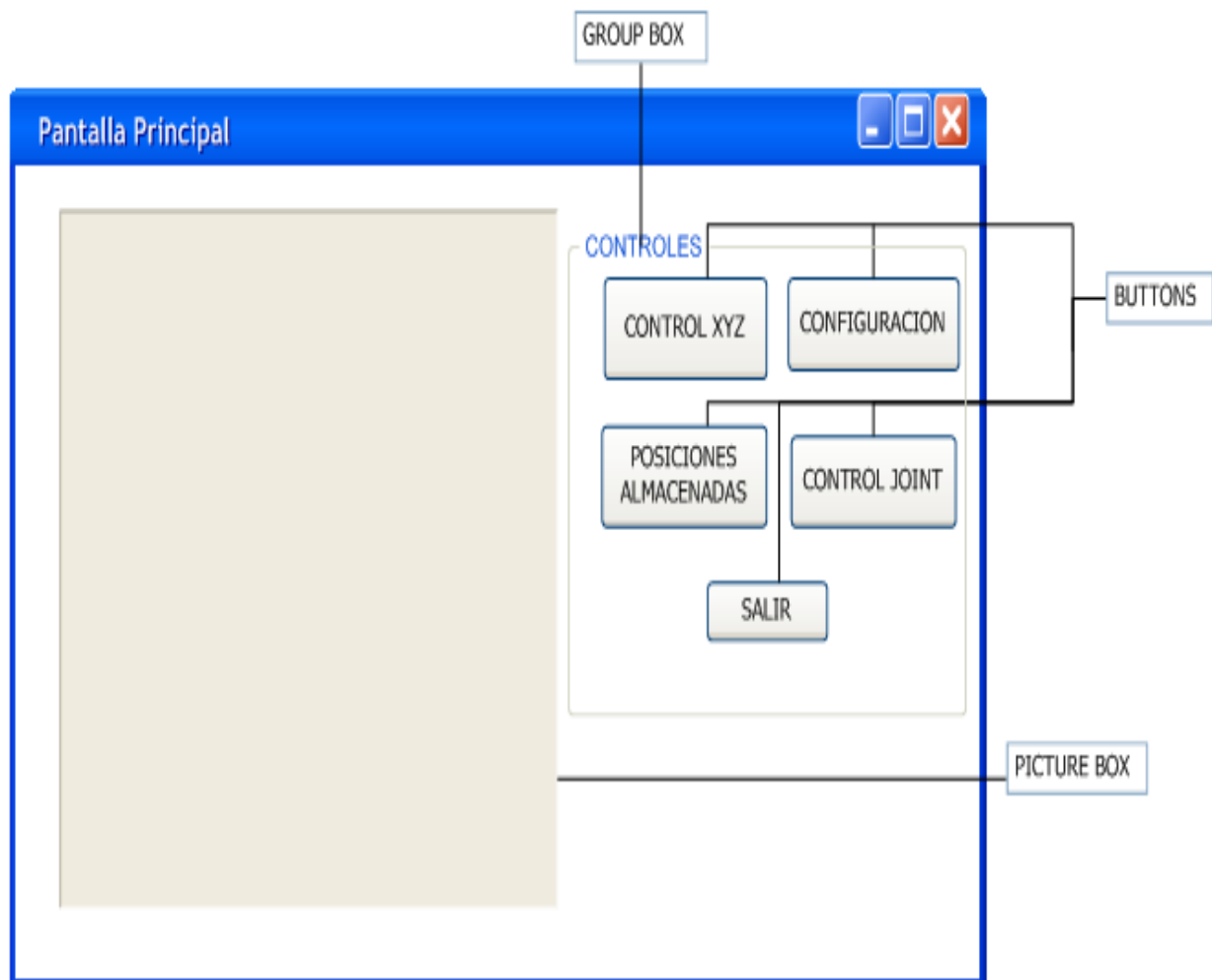


Figura 3.3.1.1 “PANTALLA PRINCIPAL DEL SISTEMA”<sup>106</sup>

<sup>106</sup> Fuente los Autores

### 3.3.2 PANTALLA MOVIMIENTO JOINT

Esta pantalla permitirá establecer los movimientos del Brazo RV-2AJ en función de los ángulos en cada junta, mediante los botones (Buttons).

Se definen parámetros como velocidad, ángulo de desplazamiento.

Los botones (Buttons2) permiten restablecer la comunicación con la controladora y salir.

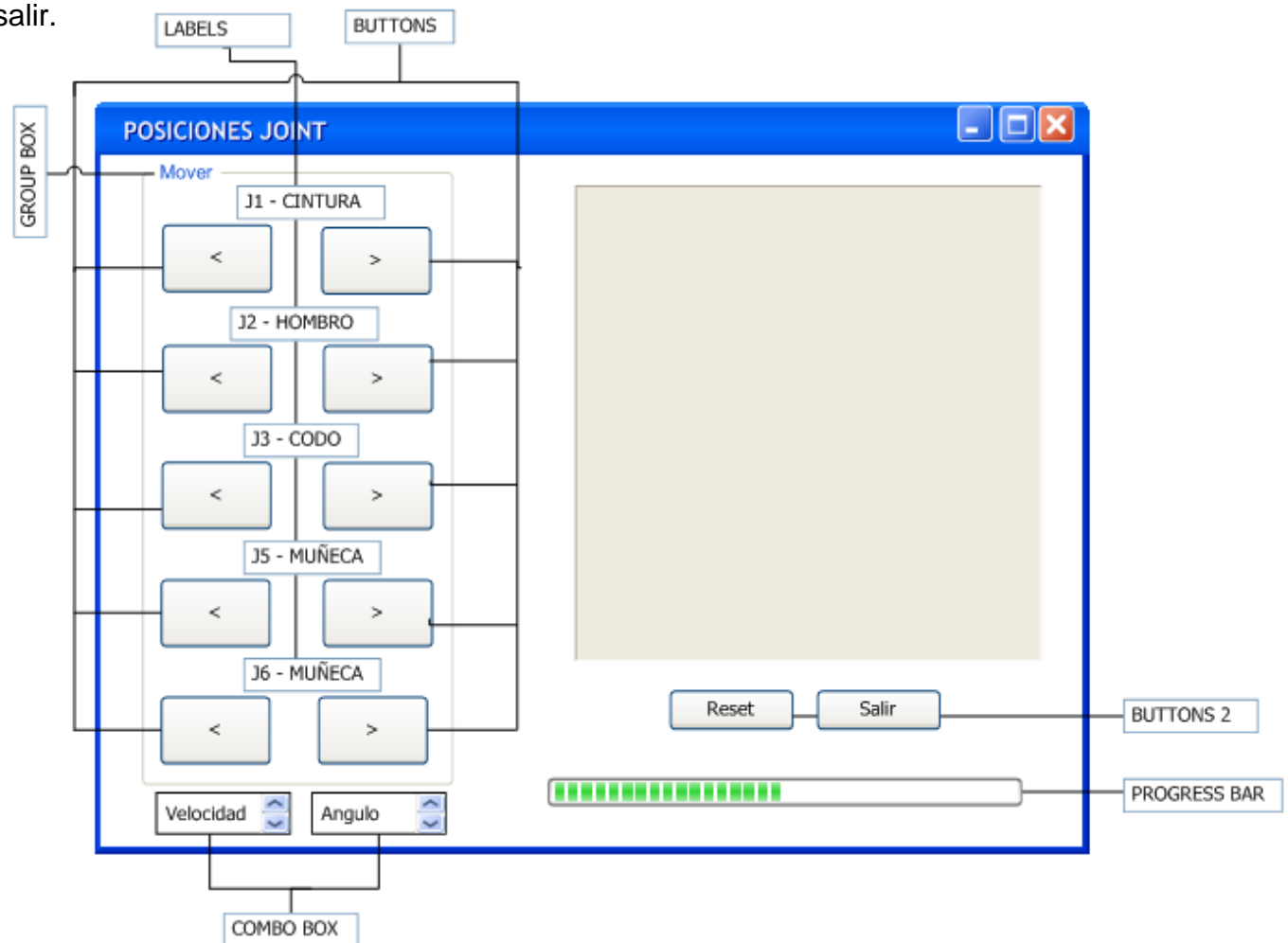


Figura 3.3.2.1 "POSICIONES JOINT"<sup>107</sup>

<sup>107</sup> Fuente los Autores

### 3.3.3 PANTALLA MOVIMIENTO POR POSICIONES ALMACENADAS

En esta ventana se procesan los programas que están almacenados en el controlador para luego proceder a ejecutar las posiciones preestablecidas dentro de cada programa.

Se accede mediante comandos configurados por default y se controla mediante hilos establecidos en la clase.

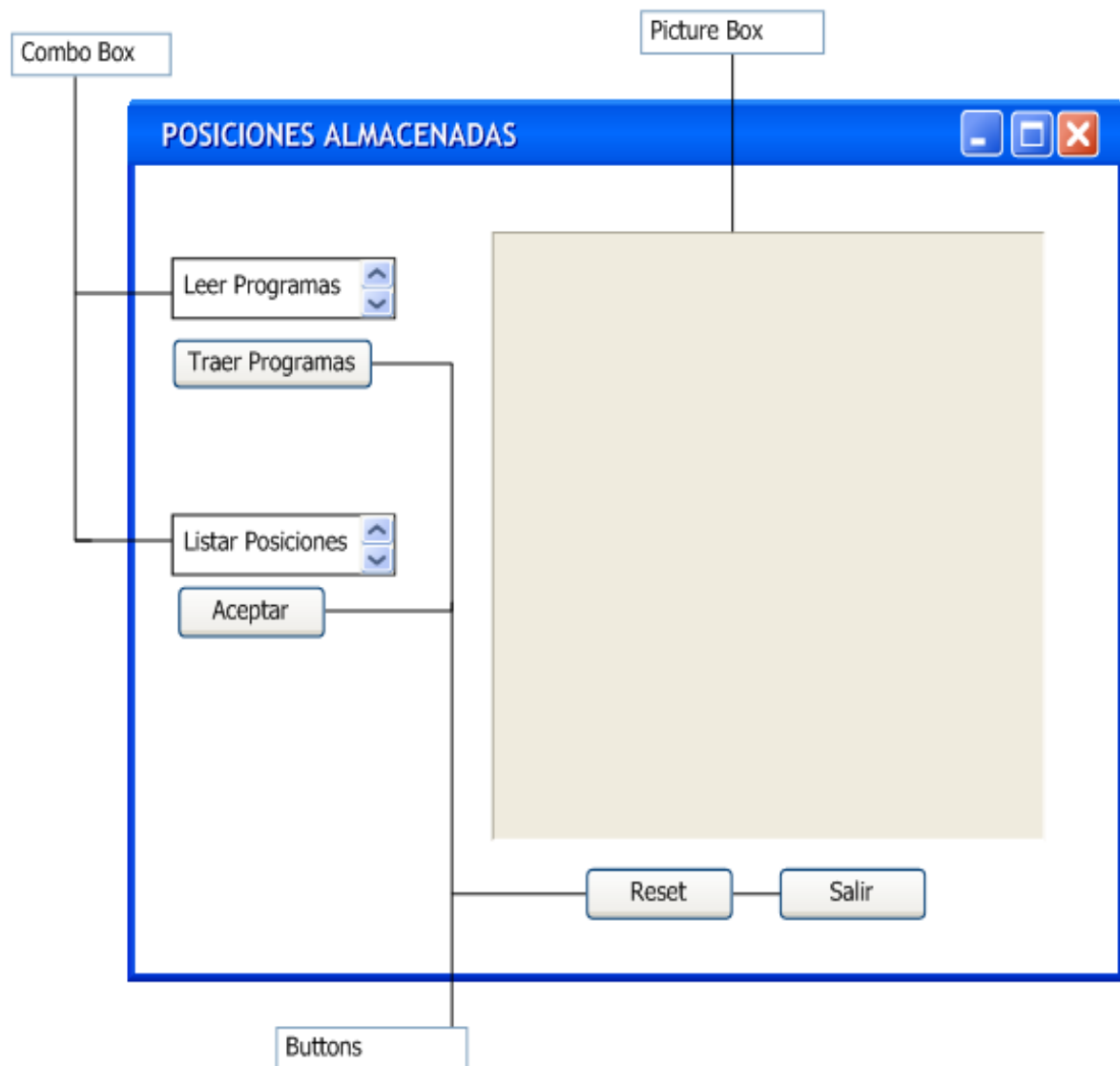


Figura 3.3.3.1 "PANTALLA POSICIONES ALMACENADAS"<sup>108</sup>

<sup>108</sup> Fuente los Autores

### 3.3.4 PANTALLA CONFIGURACIONES

En la pantalla de configuración se establecen los parámetros de configuración para generar la comunicación con la controladora.

Es importante saber que los parámetros que se establecen en esta ventana son almacenados y encriptados en un archivo para evitar cambios o pérdida de datos que cambiarían o incluso dañarían la comunicación con el brazo RV-2AJ

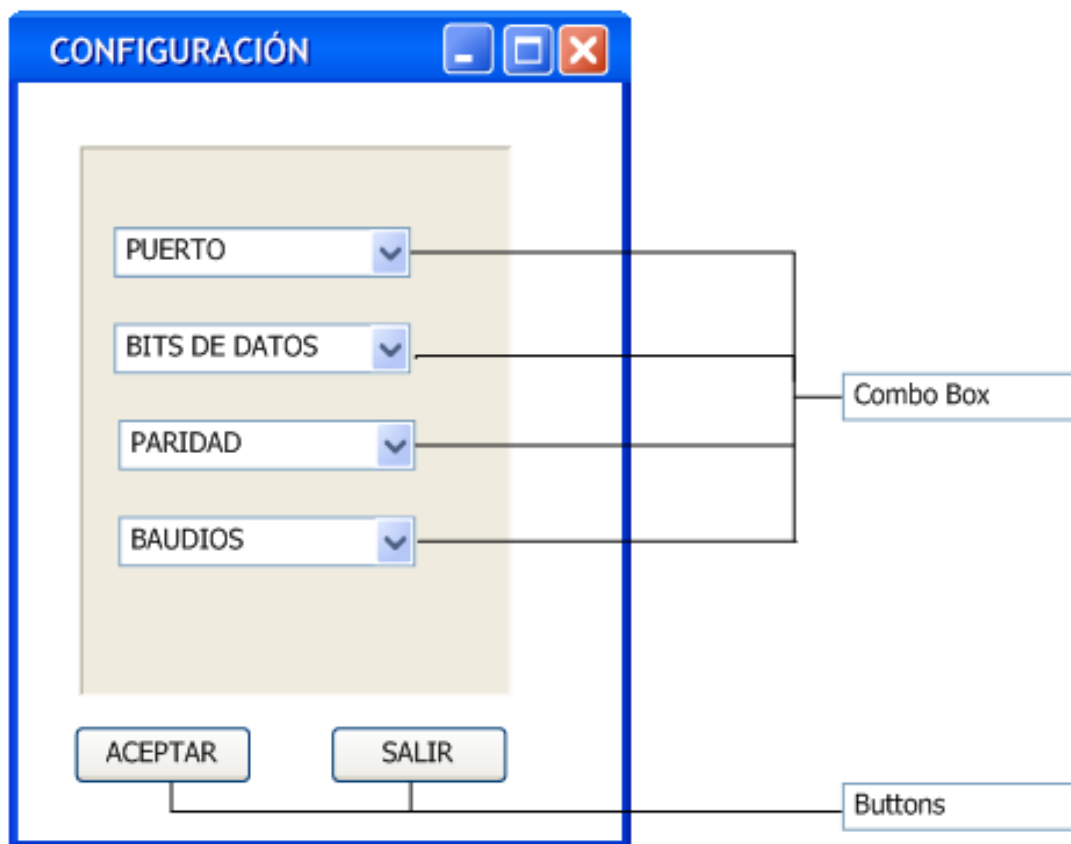


Figura 3.3.4.1 "PANTALLA CONFIGURACIONES DE COMUNICACIÓN DEL SISTEMA"<sup>109</sup>

<sup>109</sup> Fuente los Autores

### 3.4 PROGRAMACIÓN

La familia del robot de Mitsubishi incluye los robots pequeños, como el RV-2AJ que tiene una altura de 410 mm cuando el brazo se alinea verticalmente.

Se puede mencionar que el robot RV-2AJ es uno de los más conocidos a nivel didáctico educativo. El robot RV-2AJ utiliza un brazo articulado vertical de 5 ejes.

#### 3.4.1 LENGUAJE DE PROGRAMACIÓN MELFA BASIC IV

En este lenguaje la programación se estructura como un conjunto de instrucciones cuyo flujo de proceso se realiza en un lenguaje BASIC estándar, dentro del aspecto de un programa se tiene a un conjunto de instrucciones propias del sistema de Robot entre sentencias ya conocidas de BASIC.

Se obtiene así una forma intuitiva de programación, sencilla incluso para aquellos usuarios con pocos conocimientos de BASIC.

##### 3.4.1.1 INSTRUCCIONES: MOV, MVS, MVR

MOV <punto solicitado>: mueve el centro del efector final, es decir, mueve todas las articulaciones mediante la interpolación por articulaciones, desde el punto actual al punto solicitado, ejemplo:

```
10 MOV P1 ;mueve hacia P1
20 MOV P2,-50 ;mueve respecto P2, 50mm atrás de la
;posición de la mano (desp. relativo)
30 MOV P2 ;mueve hacia P2
40 MOV P3,-100 WTH M_OUT(17)=1 ;mueve respecto P3,100mm atrás, mientras
activa
;salida bit nº 17
50 MOV P3 ;mueve hacia P3
60 MOV P3,-100 ;mueve respecto P3, 100mm atrás(desp. relativo)
```

70 END ;fin de programa

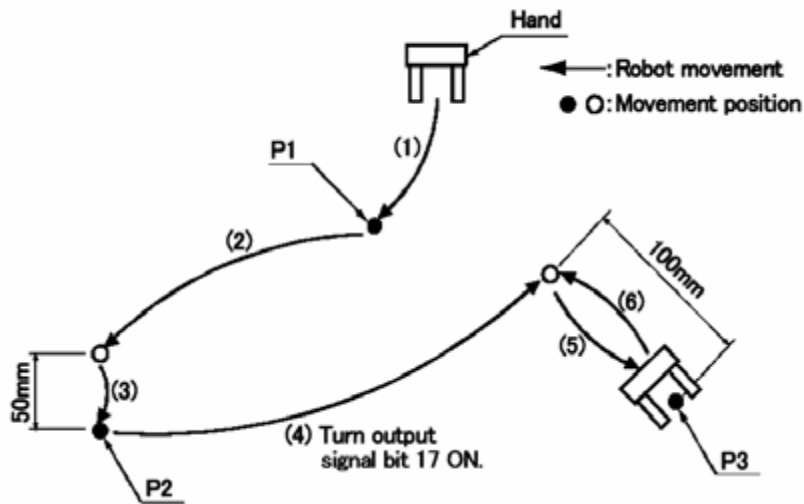


Figura 3.4.1.1.1 "Función MOV" <sup>110</sup>

MVS <punto solicitado>: mueve el centro del efector final, es decir, mueve todas las articulaciones necesarias mediante la interpolación lineal, desde el punto actual al punto solicitado siguiendo una línea recta, ejemplo:

10 MVS P1,-50 ;mueve respecto P2, 50mm atrás, en línea recta  
 ;(desp. relativo)  
 20 MVS P1 ;mueve hacia P1 en línea recta  
 30 MVS,-50 ;mueve 50mm atrás desde la posición actual  
 ; en línea recta (desp. relativo)  
 40 MVS P2,-100 WTH M\_OUT(17)=1 ;mueve respecto P2,en línea recta,  
 ;100mm atrás, mientras activa  
 ;salida bit nº 17  
 50 MVS P2 ;mueve hacia P2, en línea recta  
 60 MVS,-50 ;mueve respecto P2, en línea recta 50mm  
 ;atrás (desp. relativo)  
 70 END ;fin de programa

<sup>110</sup> Fuente CD FESTO Brazo RV-2AJ

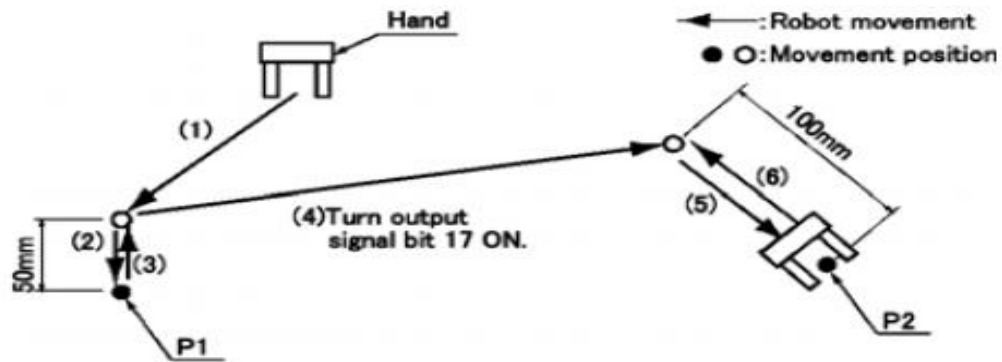


Figura 3.4.1.1.2 “Función MVS”<sup>111</sup>

MVR <punto solicitado> <punto intermedio><punto final>: mueve el centro del efector final, es decir, mueve todas las articulaciones necesarias mediante la interpolación circular, desde el punto actual P1 al punto solicitado P3 siguiendo un círculo, ejemplo:

*MVR3 P1, P2, P3*

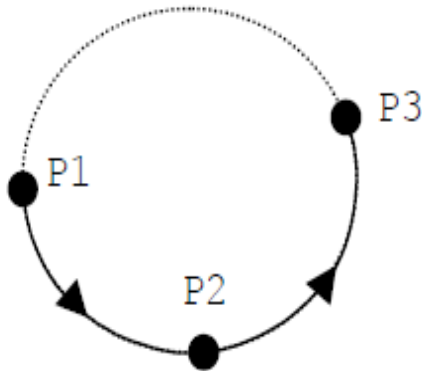


Figura 3.4.1.1.3 “Función MVR”<sup>112</sup>

<sup>111</sup> Fuente CD FESTO Brazo RV-2AJ

<sup>112</sup> Fuente CD FESTO Brazo RV-2AJ



### 3.4.2 PROTOCOLO RS-232

El puerto serial es un dispositivo muy extendido y ya sean uno o dos puertos, con conector grande o pequeño, todos los equipos PC lo incorporan actualmente.

Debido a que el estándar del puerto serial se mantiene desde hace muchos años, la institución de normalización americana (EIA) ha escrito la norma RS-232-C que regula el protocolo de la transmisión de datos, el cableado, las señales eléctricas y los conectores en los que debe basarse una conexión RS-232.

La comunicación realizada con el puerto serial es una comunicación asíncrona. Para la sincronización de una comunicación se precisa siempre de un bit adicional a través del cual el emisor y el receptor intercambian la señal del pulso. Pero en la transmisión serial a través de un cable de dos líneas esto no es posible ya que ambas están ocupadas por los datos y la tierra. Por este motivo se intercalan antes y después de los datos de información de estado según el protocolo RS-232. Esta información es determinada por el emisor y el receptor al estructurar la conexión mediante la correspondiente programación de sus puertos seriales. Esta información puede ser la siguiente:

i. Bit de inicio.- 1

Cuando el receptor detecta el bit de inicio sabe que la transmisión ha comenzado y es a partir de entonces que debe leer la transmisión y entonces debe leer las señales de la línea a distancias concretas de tiempo, en función de la velocidad determinada.

ii. Bit de parada.- 2

Indica la finalización de la transmisión de una palabra de datos. El protocolo de transmisión de datos permite 1, 1.5 y 2 bits de parada.

iii. Bit de paridad.- Impar

Con este bit se pueden descubrir errores en la transmisión.

Se puede dar paridad par o impar. En la paridad par, por ejemplo, la palabra de datos a transmitir se completa con el bit de paridad de manera que el número de bits es 1 enviados es par BaudRate 9600 de velocidad y control del Puerto Serial.

El ordenador controla el puerto serial mediante un circuito integrado específico, llamado UART (Transmisor-Receptor-Asíncrono Universal). Normalmente se utilizan los siguientes modelos de este chip: 8250 (bastante antiguo, con fallos, solo llega a 9600 baudios), 16450 (versión corregida del 8250, llega hasta 115200 baudios) y 16550A (con buffers de E/S). A partir de la gama Pentium, la circuitería UART de las placas base son todas de alta velocidad, es decir UART 16550A. De hecho, la mayoría de los módems conectables a puerto serial necesitan dicho tipo de UART, incluso algunos juegos para jugar en red a través del puerto serial necesitan de este tipo de puerto serial.

### **3.4.3 DESAFÍOS DEL MERCADO**

A nivel laboral el uso de esta práctica herramienta permite tener un manejo de sensores y actuadores a nivel general, pero teniendo ya esta noción se puede generar una integración de aplicaciones a nivel de software y hardware, teniendo un conocimiento más amplio y competitivo.

## 4 CONSTRUCCIÓN E IMPLEMENTACIÓN

Se presenta a continuación una explicación de los comandos y código para dar un seguimiento más coherente a las opciones que definen la programación, tanto del sistema como del Brazo RV-2AJ.

### 4.1 PROGRAMACIÓN DEL SOFTWARE

El API se desarrolló completamente con software libre, con el IDE .net 2010, que es una herramienta para programar en C#.

Se van a destacar algunas definiciones sobre las instrucciones que se utilizan para la programación del sistema y así tener un mejor entendimiento de los comandos y a continuación de esto se expondrán los métodos específicos.

- **BLOQUEAR / LIBERAR SERVOS**

Permiten quitar / poner los seguros mecánicos de los servos, de todas las articulaciones del brazo robótico RV-2AJ.

- **ABRIR / CERRAR PINZA**

Enciende las electroválvulas que permiten abrir o cerrar la pinza del brazo robótico.

- **DEFINIR ÁNGULO Y VELOCIDAD DE DESPLAZAMIENTO**

Permite definir el porcentaje de velocidad (0% – 100%) para desplazamientos lineales y el ángulo de desplazamiento (0° - 45°) para desplazamientos angulares.

- **POSICIONES**

Permite desplazarse entre las posiciones del programa seleccionado. Se selecciona la posición y se presiona el botón.

- **MOVER POR ARTICULACIÓN**

Permite mover el brazo, articulación por articulación, en sentido horario, anti horario y de arriba hacia abajo, según corresponda. El ángulo de desplazamiento será el definido en el GroupBox de Seteo de velocidad y ángulo de desplazamiento.

- **MOVER EN XYZ**

Permite mover el brazo en los ejes XYZ, tomando como punto 0, 0,0 el centro de la base del brazo robótico. La velocidad de desplazamiento será la definida en el GroupBox de Velocidad (mm/s) e Incremento (mm). El uno define cuántos milímetros recorrerá por segundo y el otro define cuántos milímetros de desplazará por pulso, ya que el brazo en modo teach no tiene un movimiento continuo.

- **RESET**

Es el equivalente en software, al botón de reset que existe en el teach y en la controladora. Resetea las alarmas activadas y permite seguir manipulando el brazo robótico.

## 4.2 MÉTODOS PARA MOVER EL BRAZO

### 4.2.1 CONDICIONES DE INICIO

*Void MainFormLoad (object sender, EventArgs e)*

```
{
    RV_2AJ.set_par("COM5");
    RV_2AJ.abrir_com();
    leer_XYZ();
}
```

Define el puerto por el cual se va a realizar la comunicación, abre el puerto y hace una petición a la controladora para saber en qué posición XYZ se encuentra la pinza en relación a la base del brazo robótico.

### 4.2.2 MOVER EN XYZ

*void mover\_XYZ()*

```
{
    RV_2AJ.mover_XYZ((double)txtVelocidadXYZ.Value,
    (double)txtX.Value, (double)txtY.Value, (double)txtZ.Value,
    (double)txtA.Value, (double)txtB.Value); }
```

Llama al método mover\_XYZ de la clase clsRV-2AJ, que recibe como parámetros:

- **Parámetro 1:** Desplazamiento en milímetros por segundo.
- **Parámetro 2:** Valor a desplazarse en el eje X
- **Parámetro 3:** Valor a desplazarse en el eje Y
- **Parámetro 4:** Valor a desplazarse en el eje Z
- **Parámetro 5:** Valor a desplazarse de una articulación de la muñeca
- **Parámetro 6:** Valor a desplazarse de otra articulación de la muñeca

#### 4.2.3 MOVER POR ARTICULACIÓN

```
void BtnJ1Click(object sender, EventArgs e)
```

```
{
```

```
RV_2AJ.mover_articulacion
```

```
(-1, 1, (double)txtVelocidad.Value, (double)txtAngulo.Value);}
```

Llama al método mover\_articulación de la clase clsRV-2AJ, que recibe como parámetros:

- **Parámetro 1:** sentido de giro, 1 para horario y subir, y -1 para anti horario y bajar.
- **Parámetro 2:** articulación a mover, de la 1 a la 6, excluyendo la número 4 que corresponde a una articulación que no tiene el modelo RV-2AJ.
- **Parámetro 3:** velocidad de movimiento en porcentaje del 0% al 100%.
- **Parámetro 4:** valor del ángulo que se va a desplazar la articulación.

El método mostrado permite mover la articulación J1 que corresponde a la cintura, si se desea mover las demás articulaciones solamente es necesario cambiar el segundo parámetro por el número de articulación según la tabla.

Designación	Articulación
J1	Cintura
J2	Hombro
J3	Codo
J4	No existe en RV-2AJ
J5	Pitch <sup>113</sup> Muñeca
J6	Roll <sup>114</sup> Muñeca

Tabla 4.2.3.1 Designación de articulaciones del brazo RV-2AJ<sup>115</sup>

#### 4.3 LEER LOS PROGRAMAS ALMACENADOS EN LA MEMORIA DE LA CONTROLADORA

```
void BtnLeerProgramasClick(object sender, EventArgs){
    string [] progs = RV_2AJ.leer_programas();

    cbProgramas.Items.Clear();

    for (int i = 0; i < progs.Length; i++)
    {

        if (!progs[i].Contains("Q"))cbProgramas.Items.Add(progs[i]); }

    cbProgramas.Text = cbProgramas.Items[0].ToString();}
```

Llama al método leer\_programas de la clase clsRV-2AJ, que devuelve los nombres de todos los programas almacenados en la memoria de la controladora, como un vector tipo String, que son llenados en un combo para escoger que lista de posiciones de que programa se quiere recuperar.

<sup>113</sup> Pitch: giro de la pinza en el eje Y, también llamado balanceo o cabeceo.

<sup>114</sup> Roll: giro de la pinza en el eje X, también llamado rotación.

<sup>115</sup> COSIROP

#### 4.4 CLASE PARA COMUNICACIÓN E INSTRUCCIONES DE MOVIMIENTO

Se realizó una clase que se encargará de la comunicación vía serial del programa con la controladora del brazo robótico RV-2AJ.

##### 4.4.1 MÉTODO PARA DEFINICIÓN DE PARÁMETROS DE COMUNICACIÓN

```
public void set_par(string puerto)
{
    com.PortName = puerto;

    com.BaudRate = 9600;

    com.Parity = Parity.Even;

    com.StopBits = StopBits.Two;

    com.DataBits = 8;
}
```

Este es el método más importante, ya que define los parámetros a los cuales se va a realizar la comunicación.

##### 4.4.2 MÉTODO PARA ENVÍO DE DATOS A LA CONTROLADORA

```
public void escribir(string instruccion)
{
    com.Write("1;1;" + instruccion + "\r");

    Thread.Sleep(10);
}
```

Permite enviar los datos con la sintaxis correcta para que la controladora entienda las órdenes. La sintaxis es la siguiente

- **Parámetro “1;1;”:** define que se enviarán instrucciones de movimiento.
- **Instrucción:** aquí va cualquier dato que se quiera enviar a la controladora.
- **Parámetro “\r”:** es necesario para finalizar la línea enviada.

#### 4.4.3 MÉTODO PARA ENVÍO DE COMANDOS A LA CONTROLADORA

```
public void comando(string comando)
```

```
{
    escribir("CNTLON");
    escribir(comando);
    escribir("CNTLOFF");
}
```

Permite enviar comandos de movimiento a la controladora para que se mueva el brazo. La sintaxis es la siguiente:

- **Instrucción CNTLON:** define que se enviará un dato de control sobre el brazo y la inicializa.
- **Comando:** es el comando de movimiento que se enviará a la controladora.
- **Instrucción CNTLON:** finaliza la etapa de envío de instrucciones de control sobre el brazo.

#### 4.4.4 MÉTODOS PARA ACTIVACIÓN Y DESACTIVACIÓN DE SERVOS

```
public void servo_on()
```

```
{
    comando("EXECSEVO ON");
}
```

```
public void servo_off()
```

```
{
```



```

    comando("EXECSEVO OFF");

}

```

Usando el método comando(), permite activar y desactivar los servos del brazo. Se utilizan las mismas órdenes que en MELFA-BASIC, con la diferencia que hay que adicionarle "EXEC" al inicio del comando, para que se convierta en una orden de ejecución y no dependa de un programa para realizarse.

#### 4.4.5 MÉTODOS PARA ABRIR Y CERRAR GRIPPER

```

public void abrir_gripper(){

    comando("EXECHOPEN 1");}

public void cerrar_gripper()

{

comando("EXEHCLOSE 1");

}

```

Usando el método comando(), y la orden "EXECHOPEN 1" se puede abrir y cerrar el gripper del brazo. Se utilizan las mismas órdenes que en MELFA-BASIC, con la diferencia que hay que adicionarle "EXEC" al inicio del comando, para que se convierta en una orden de ejecución y no dependa de un programa para realizarse.

#### 4.4.6 MÉTODOS PARA DESPLAZARSE A UNA POSICIÓN DETERMINADA

```

public void mover_a_pos(int pos)

{
    if (pos < 10)
    {

        comando("EXECMOV P0" + pos);
    }

    Else
    {

        comando("EXECMOV P" + pos);

    }
}

```

Usando el método comando(), y la orden “EXECMOV P + pos” se puede desplazar el brazo una posición guardada en el positionList almacenado en la controladora. Se utilizan las mismas órdenes que en MELFA-BASIC, con la diferencia que hay que adicionarle “EXEC” al inicio del comando, para que se convierta en una orden de ejecución y no dependa de un programa para realizarse.

## 4.5 PROGRAMACIÓN DEL ROBOT

El API realizado en esta tesis no realiza ningún tipo de programación directa al robot, pero si utiliza el lenguaje MELFA – BASIC, para el envío de las órdenes al robot y manipularlo desde una aplicación libre vía Serial.

Los comandos utilizados para controlar el robot son:

- CNTL (control): Designa el control para movimientos suaves entre desplazamientos de interpolación<sup>116</sup>.
- CNTLON: Palabra reservada que habilita el control para movimientos suaves entre desplazamientos de interpolación.
- CNTLOFF: Palabra reservada que deshabilita el control para movimientos suaves entre desplazamientos de interpolación.
- PPOSF: Palabra reservada que envía un comando de consulta al robot sobre su posición actual, en función de X, Y, Z, A y B<sup>117</sup>. La controladora devuelve un string con las posiciones en ese orden.
- MOV (move): Usando una operación interpolaciones de articulaciones, el brazo se mueve de la posición actual a la posición de destino.

---

<sup>116</sup> En el subcampo matemático del análisis numérico, se denomina interpolación a la obtención de nuevos puntos partiendo del conocimiento de un conjunto discreto de puntos.

<sup>117</sup> Hace referencia a la posición X, Y, Z, A, B, del centro de la pinza tomando como origen el punto 0, 0, 0, 0,0 que se encuentra en la base del brazo robot.

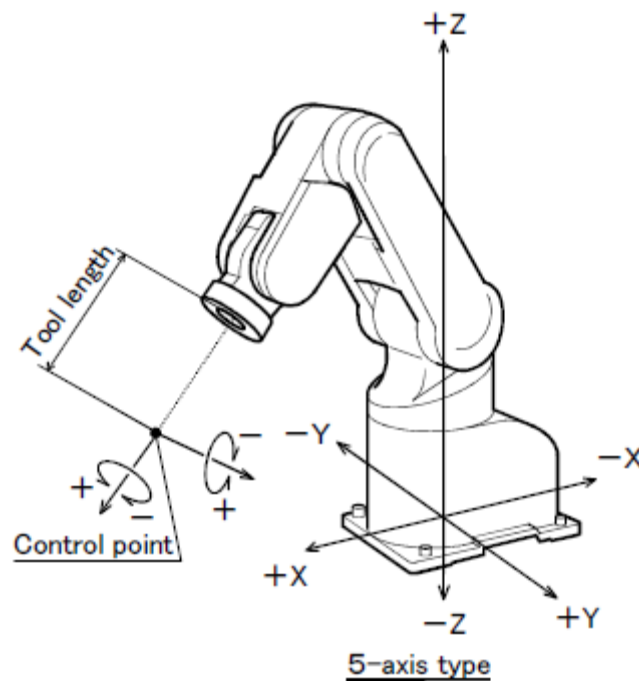


Figura 4.5.1 “Desplazamientos ROBOT RV-2AJ”<sup>118</sup>

- EXECMOV: Palabra reservada que permite mover el brazo a una posición determinada dentro de la lista de posiciones del programa. El formato es:  
EXECMOV P#  
Si el número es menor a 10 hay que enviar el número de la posición precedido de un 0 (cero) .
- RSTALRM (reset alarm): Resetea todas las alarmas generadas por el robot, sean estas por problemas de comunicación, colisión, programación, etc.
- SPD (speed): Designa la velocidad del robot para movimientos lineales y circulares.
- EXECSPD: Palabra reservada que setea la velocidad del robot para movimientos lineales y circulares.
- EXECPCOSIROP: Palabra reservada que permite setear las posiciones finales para mover el robot en X, Y, Z desde la posición actual. El formato es:  
(X, Y, Z, A, B, 0.00) (6,0))

<sup>118</sup> Fuente CD FESTO Brazo RV-2AJ

- MVS (move S): Lleva a cabo un movimiento de interpolación lineal desde la posición actual a la posición de destino.
- EXECMVS PCOSIROP: Palabra reservada que lleva a cabo un movimiento de interpolación lineal desde la posición actual a la posición de destino, desde el API en C# a través del puerto serial.

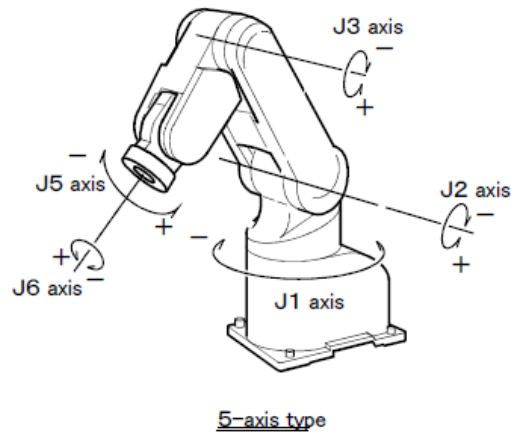


Figura 4.5.2 Desplazamientos de las articulaciones ROBOT RV-2AJ<sup>119</sup>

- JOVRD (J override): Designa el único comando de anulación que es válido durante el movimiento por articulación del robot.
- EXECJOVRD: Palabra reservada utilizada para setear la velocidad a cual se moverá el robot, en interpolaciones para movimientos de articulaciones.
- EXECJCOSIROP: Palabra reservada que permite setear los parámetros para mover el robot por articulación. El formato es:  

$$\text{EXECJCOSIROP} = (X, Y, Z, A, B, C)$$

$$\text{EXECJCOSIROP} = (0.000, 0.000, 0.000, 0.000, 0.000, 0.000)$$
- J\_CURR: Regresa el dato tipo articulación de la posición actual.
- EXECMOV J\_CURR + JCOSIROP: Envío de dos palabras reservadas que devuelven la posición actual en la que se encuentra el robot en modo Joint y ejecutan el movimiento de las articulaciones desde la posición actual hasta la posición de destino.

<sup>119</sup> Fuente CD FESTO Brazo RV-2AJ

- SERVO: Controla el encendido y apagado de los servos añadiendo ON u OFF a continuación de SERVO.
- EXECSERVO ON: Palabra reservada que envía el comando para encender los servos.
- EXECSERVO OFF: Palabra reservada que envía el comando para apagar los servos.
- HOPEN/HCLOSE (Hand Open/Close): Controla la apertura y cierre de la pinza.
- EXECHOPEN / EXECLOSE: Palabra reservada que permite el envío del comando para poder abrir y cerrar la pinza.

#### 4.6 RESUMEN DE PROTOCOLOS Y COMANDOS Y CONTENIDOS.

- La comunicación con el robot se realiza a través de RS-232. Para enviar cualquier comando es necesario enviar 2 números separados por punto y coma. Esto representa un código que le indica a la controladora que tipo de instrucción va a recibir. En este proyecto se utilizan 2 tipos de comandos con el siguiente formato:

**##; instrucción \r**

- 1;1;** estos 2 números al inicio de la instrucción indican a la controladora que se va a enviar un comando de movimiento.
- 1;9;** estos 2 números al inicio de la instrucción indican a la controladora que se va a enviar un comando de consulta de programa.

Al final de la línea siempre se debe enviar un salto de línea \r como carácter de finalización de instrucción.

La siguiente tabla describe las instrucciones enviadas al robot.

PROTOCOLO DE COMUNICACIÓN PC -> CONTROLADORA		
SENTENCIA	ACCIÓN	PARÁMETRO
CNTLON	Inicia sentencia de control	
CNTLOFF	Finaliza sentencia de control	
PPOSF	Envía petición de posicionamiento	
RSTALRM	Resetea las alarmas activadas	
EXECPCOSIROP	Define parámetros para realizar un movimiento en X,Y,Z,A,B	("EXECPCOSIROP = (" + X + ", " + Y + ", " + Z + ", " + A + ", " + B + ",0.00)(6,0)");
EXECMVS PCOSIROP	Ejecuta movimiento en XYZAB	

EXECJCOSIROP	Define parámetros para realizar movimiento por articulación	("EXECJCOSIROP = ("0.000, 0.000, 0.000, 0.000, 0.000)");
EXECMOV J_CURR + JCOSIROP	Ejecuta movimientos por articulación con los parámetros anteriores	
EXECSERVO ON	Desbloquea servos	
EXECSERVO OFF	Bloquea servos	
EXECHOPEN 1	Abre gripper	
EXECHCLOSE 1	Cierra gripper	
EXECMOV P	Desplaza el brazo a la posición guardada en memoria	"EXECMOV P" + pos
PDIR	Envía petición de lista de programas	
PRTVERLISTL	Envía petición de lista de posiciones	
LOAD	Carga en memoria el programa seleccionado	
1;9;LIST< \r	Indica que las posiciones deben ser enviadas como lista	
1;9;VALS<;P \r	Sentencia especial para cargar en la variable cada posición	
EXECJOVRD	Setea la velocidad para moverse por articulación	"EXECJOVRD " + vel
EXECSPD	Setea la velocidad para moverse en XYZAB	"EXECSPD " + vel

Tabla 4.6.1 Instrucciones del robot<sup>120</sup>

<sup>120</sup> COSIROP

## 4.7 INTERFAZ FINAL DEL SOFTWARE

Por medio de las cuales se dan a conocer de manera gráfica todas las pantallas de manejo del brazo se desarrollaron para el sistema, además se da una descripción de su uso y los botones o eventos que se van a generar en la corrida del programa.

### 4.7.1 PANTALLA PRINCIPAL

Es el menú principal del programa desde donde se puede acceder a todas las funcionalidades del API.

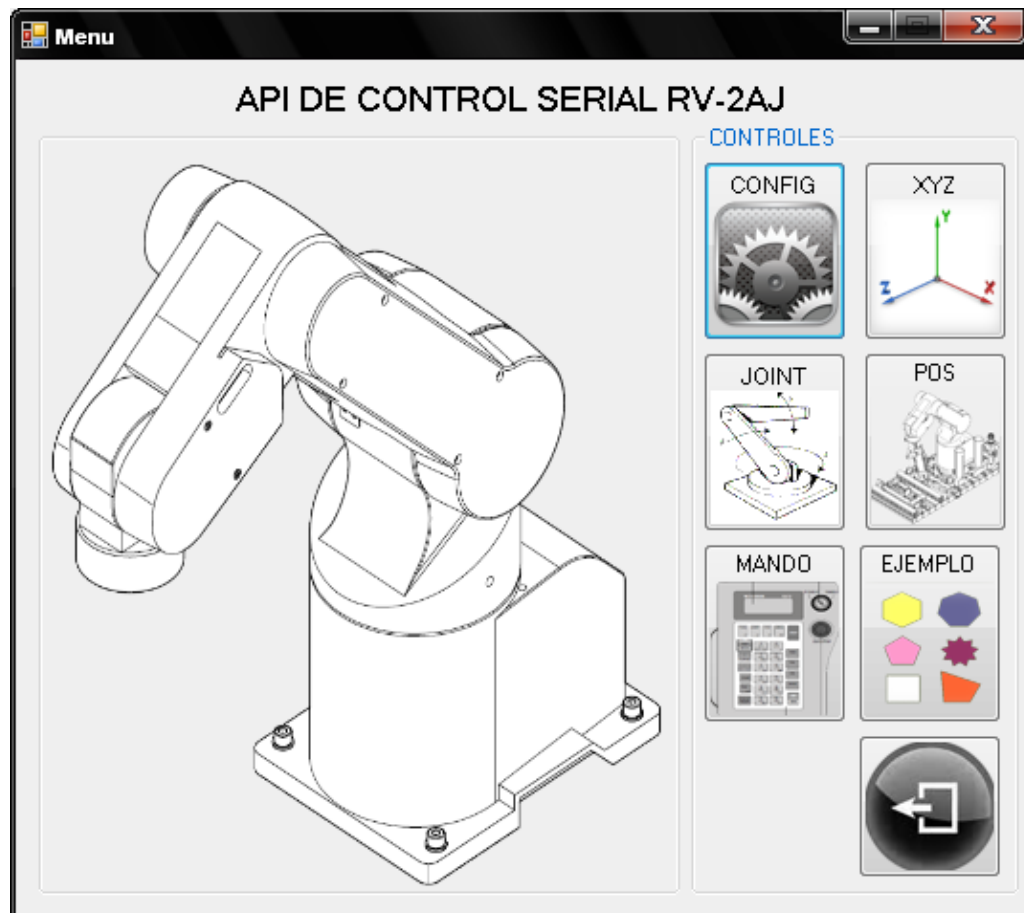


Figura 4.7.1.1 Menú principal del proyecto <sup>121</sup>

<sup>121</sup> Fuente los Autores



## 4.7.2 PANTALLA DE CONFIGURACIÓN

En esta pantalla se definen los parámetros para la comunicación serial.



Figura 4.7.2.1 Configuración comunicación con el proyecto<sup>122</sup>

## 4.7.3 PANTALLA DE MOVIMIENTO EN MODO XYZ

Esta pantalla permite mover el brazo en los planos XYZ AB, abrir / cerrar el gripper y moverse a una posición específica en XYZ AB.

---

<sup>122</sup> Fuente los Autores

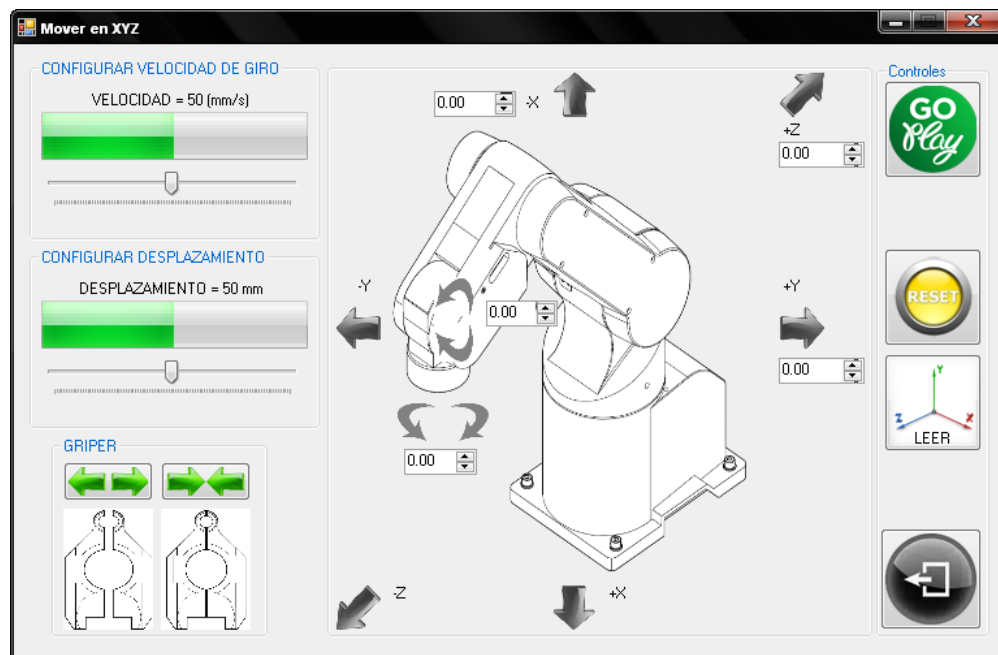


Figura 4.7.3.1 Pantalla de desplazamientos en modo XYZ<sup>123</sup>

#### 4.7.4 PANTALLA DE MOVIMIENTO POR ARTICULACIONES

Esta pantalla permite mover el brazo por articulaciones desde la cintura hasta el efector final.

<sup>123</sup> Fuente los Autores

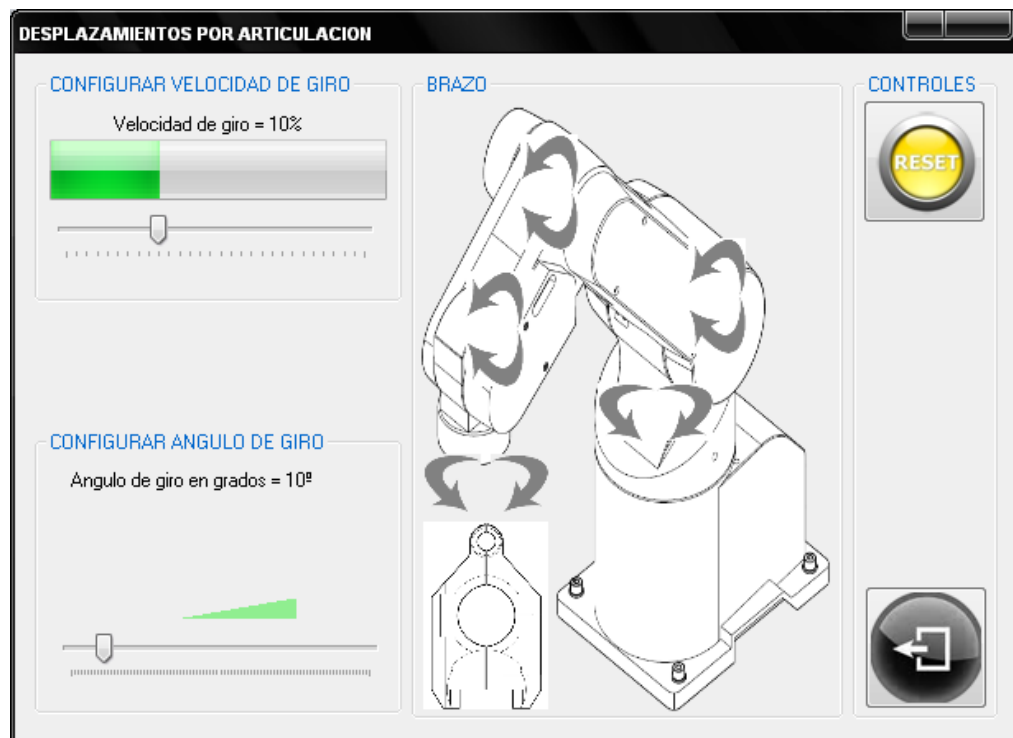


Figura 4.7.4.1 Pantalla de desplazamientos en modo JOINT<sup>124</sup>

#### 4.7.5 PANTALLA DE MOVIMIENTO POR POSICIONES

Esta pantalla permite moverse entre las posiciones del programa seleccionado.

<sup>124</sup> Fuente los Autores

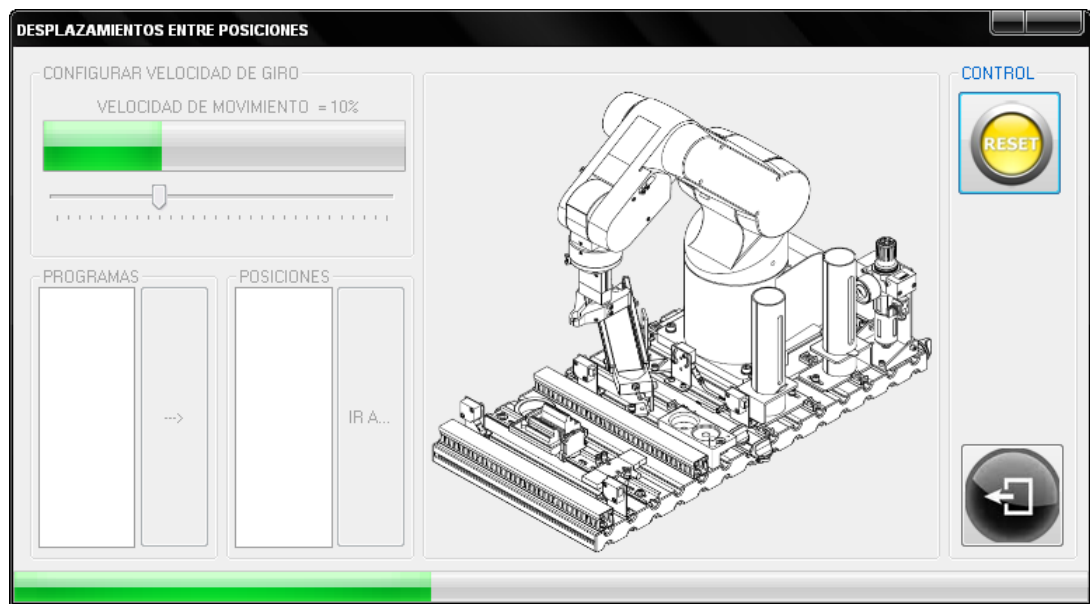


Figura 4.7.5.1 Pantalla de carga de programas y posiciones <sup>125</sup>

#### 4.7.6 TEACH

Esta pantalla permite moverse en modo XYZAB y por articulación, pero con una interfaz más compacta para operaciones más rápidas.

<sup>125</sup> Fuente los Autores



Figura 4.7.6.1 Pantalla con todas las funciones para el control del Brazo <sup>126</sup>

<sup>126</sup> Fuente los Autores

### 4.7.7 EJEMPLO

Esta es una pantalla de ejemplo donde se muestra una pequeña implementación del API, desarrollada en este proyecto.

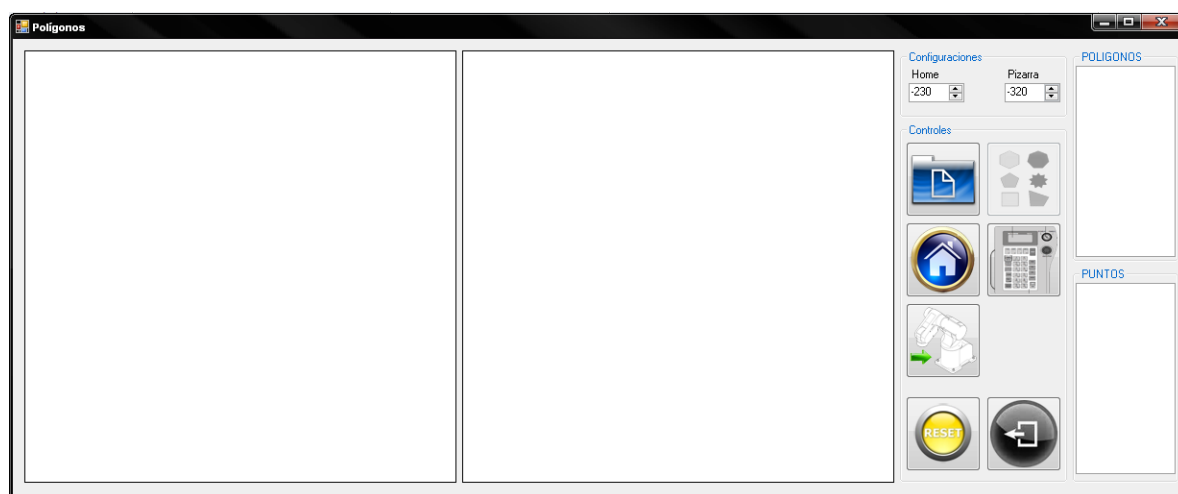


Figura 4.7.7.1 Pantalla con interfaz de ejemplo<sup>127</sup>

## 4.8 PRUEBAS Y AJUSTES

Las primeras pruebas se realizaron en un robot RV-M1<sup>128</sup> de Mitsubishi y Visual Basic 6.0 utilizando comunicación serial.

<sup>127</sup> Fuente los Autores

<sup>128</sup> El robot RV-M1 pertenece al Subcentral de Electricidad y Electrónica del CERFIN (Centro de Formación Industrial del Norte).



Figura 4.8.1 “Robot RV-M1”<sup>129</sup>

Una vez probado el funcionamiento de la comunicación con el robot RV-M1 se decidió probar el robot RV-2AJ, con el mismo programa.

Para probar que la comunicación con el robot RV-2AJ era posible se utilizó COSIROP, y la herramienta Jog Operation, para mover el brazo desde un programa externo, lo que en un inicio no dio resultado.

---

<sup>129</sup> SECAP

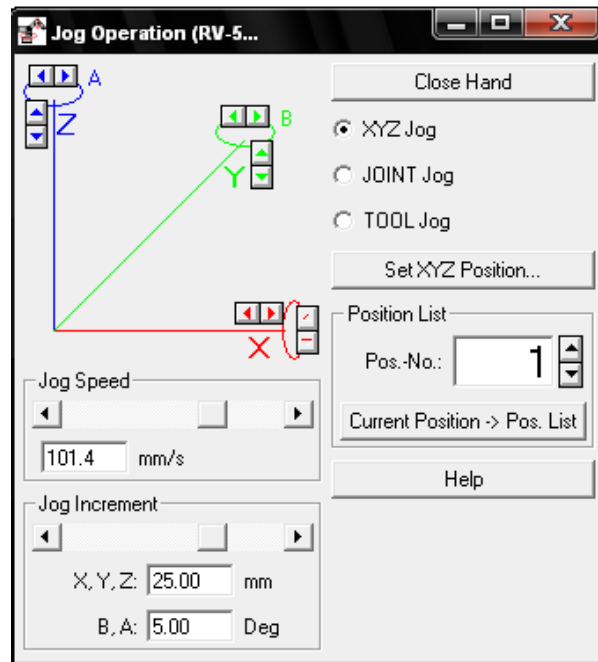


Figura 4.8.2 “Robot RV-M1” <sup>130</sup>

Se hizo pruebas básicas de comunicación serial con micro controladores y se encontró que había problemas con los drivers del adaptador USB-SERIAL, por lo que fueron reinstalados, corrigiendo el problema de envío de datos vía serial.

Se hizo nuevamente las pruebas de comunicación con el robot RV-2AJ desde COSIROP, obteniendo resultados positivos. De aquí en adelante para cada comando que se quisiera probar se realizaba una prueba con el Job Operation y luego se procedía a hacer la prueba en el IDE.

Se corrigieron los parámetros de comunicación según los definidos por COSIROP 3.0<sup>131</sup> para el robot RV-2AJ, y se probó la comunicación lo que dio un error en la controladora. Se cambió un parámetro al azar de la comunicación serial por un valor aleatorio, y se probó la comunicación, nuevamente no se obtuvo ningún resultado,

<sup>130</sup> Fuente los autores- programa Cosirop

<sup>131</sup> Es un programa de Mitsubishi anterior a CYROS, que permite leer y escribir en la controladora de los robots tipo RV.



con lo que se confirmó que con los parámetros definidos por COSIROP, si se recibía la información pero en un formato incorrecto.

Se revisó la documentación sobre el control del robot RV-2AJ por comunicación serial desde un programa externo, específicamente como abrir y cerrar la pinza, ya que aún no se tenía un buen control sobre el robot RV-2AJ y era el elemento más fácil de controlar sin mover el robot y correr riesgo de dañarlo. Se hizo pruebas con algunas variaciones de sintaxis, hasta eliminar las alarmas emitidas por la controladora. Una vez logrado esto, tampoco se consiguió mover la pinza, lo que llevó a revisar la mecánica del robot lo llevó a encontrar que la alimentación de aire de la pinza estaba desconectada.

Se conectó la alimentación de aire y se hizo nuevamente la prueba de comunicación con el robot, lo que dio un resultado positivo, al cerrar y abrir la pinza utilizando el API.

Una vez validados los parámetros y los estándares de comunicación se validaron los comandos de control del lenguaje MELFA BASIC IV.

Cuando se trató de mover el brazo en el modo XYZ, se tuvo la necesidad de saber la posición en la cual se encontraba el robot. Entonces se debió investigar la comunicación bidireccional con el brazo, ya que hasta el momento solo se había realizado el envío de datos desde la computadora hacia la controladora.

Una vez se obtuvo respuesta de la controladora la información no era la deseada ya que llegaba incompleta o fragmentada, por lo que se le dio una pausa entre el envío del comando y la lectura de la respuesta. Esto dio información más completa, pero no la deseada, lo que llevó a hacer un cálculo en el tiempo de espera, para obtener la información completa.

Una vez obtenida la posición absoluta de la pinza, se procedió a hacer incrementos o decrementos relativos a caja eje, y enviar las nuevas posiciones absolutas a la controladora, con lo que se logró un posicionamiento exacto del robot.

## **4.9 CONCLUSIONES Y RECOMENDACIONES**

Aquí se muestra las conclusiones que obtuvimos a lo largo del desarrollo del proyecto y las recomendaciones para el mejor manejo del API y el robot.

### **4.9.1 CONCLUSIONES**

- Para el mejor manejo de los métodos de comunicación, se implementaron hilos que le daban una mayor fluidez al programa, evitando pausas innecesarias en los momentos de consulta hacia y desde la controladora.
- Dependiendo del tipo de equipo que se utilice, será necesario hacer uso de un driver USB – SERIAL, si es que la computadora no posee un COM físico. Esto llevó a la configuración de los puertos COMs para cada una de las pruebas.
- Se investigó las configuraciones del protocolo propietario de MITSUBISHI, para la comunicación de la PC con el Robot. Lo que mostró que cada serie de robots tiene su propia configuración específica para funcionar y no existe un estándar para toda la gama de robots Mitsubishi.
- Al final se desarrolló una API mostrando la utilidad de la dll, logrando una mayor velocidad en el momento del desarrollo de las aplicaciones e integración con funcionalidades que el Robot RV-2AJ no poseía, como es una comunicación vía Ethernet.
- Una vez realizado el estudio del lenguaje se procedió con la comunicación con el brazo RV-M1, se pudo concluir que la comunicación con este tipo de brazos era posible, lo que motivó para seguir con la investigación, y aplicar ese conocimiento al brazo RV-2AJ.
- Fue de gran ayuda la guía del Tglo. José Zapata, quien había desarrollado un proyecto parecido sobre el brazo RV-M1, y nos permitió comparar resultados entre los 2 proyectos.
- Cuando se realiza una consulta a la controladora del robot, es necesario hacer una pausa entre el envío y la recepción, ya que si no se hace así, se recibirán

datos incompletos y no aptos para el procesamiento debido al retardo que tiene la plataforma .net al enviar los datos a la controladora.

- Dentro del programa se trabaja con incrementos relativos a la posición de la pinza, pero a la controladora se envían valores absolutos de posición en X, Y, Z, A, B.
- Para probar cada comando del programa, se utilizó una herramienta que funciona perfectamente a nivel de comunicación y sintaxis, como es el Jog Operation del programa COSIROP. Esto permitía saber que el comando funcionaba y era posible utilizarlo dentro del API.
- Utilización de las librerías en aplicaciones de visión e inteligencia artificial al poder vincular directamente los movimientos con el Brazo Robótico.
- La proyección que se genera a partir de la API permite tener un margen más amplio de aplicación, como por ejemplo:
  - Control Web: permite tener una interconexión remota vía internet
  - A\*: Permite que el sistema pueda interactuar con un aprendizaje a nivel neuronal para generar repeticiones de posiciones, no solo a nivel de programación en la controladora sino mediante procesos de auto generación de código
  - Visión Artificial que permite tener una vinculación más amplia, práctica y rápida de conexión con una cámara, para tener reconocimiento del entorno y generar aplicaciones de mayores capacidades.
- Al hacer las pruebas finales sobre los desplazamientos en XYZ, el brazo mostró cierta anomalía al momento de llegar a posiciones exactas sobre el plano Y, ya que tenía errores de varios milímetros en el eje X en relación a la posición deseada.
- Las anomalías presentadas anteriormente en el plano Y, no se presentaban en el plano Z, y si se lograba llegar a las posiciones de manera exacta.
- Para saber si el robot terminó de realizar algún movimiento o ha llegado a una posición deseada, se generó un método de consulta a la controladora, que

valida la posición actual con la posición deseada cada 100 milisegundos, porque no está disponible esta función por software.

- Una de las mejores formas de fomentar el desarrollo de nuevas aplicaciones, es abrir las herramientas de desarrollo, para que varias personas puedan trabajar sobre la misma herramienta con diferentes enfoques, es por eso que este trabajo de tesis termina con una dll compatible con la plataforma .Net.

#### **4.9.2 RECOMENDACIONES**

- Antes de iniciar cualquier tipo de manipulación sobre el robot es recomendable revisar que todas condiciones normales de funcionamiento estén dadas:
  - Las fuentes de alimentación estén habilitadas y funcionando
  - Las botones de pánico estén desenclavados
  - La llave de control esté colocada en la controladora y en modo de comunicación externa
  - Los cables de comunicación estén correctamente conectados a la controladora, al robot y la computadora
  - Los drivers de comunicación estén instalados correctamente en la computadora
- Es recomendable reducir los incrementos y las velocidades de desplazamiento, sean lineales o angulares, al momento de hacer pruebas, ya que esto nos permite reaccionar a tiempo, en caso de una posible colisión.
- Cuando se vaya a realizar algún movimiento del robot desde el programa, es siempre aconsejable sujetar el TEACH, con la disposición de presionar el botón de emergencia, a pesar de que sean movimientos ya conocidos, ya que es un equipo muy costoso y no estamos exentos a cometer errores.
- Cuando suene alguna alarma por algún movimiento indebido, se la puede desactivar desde el programa, pulsando el botón de RESET que tienen todas las interfaces, y no ir al teach o a la controladora.
- Cuando se quiera pausar un movimiento a mitad del camino, hay que presionar el botón de STOP en la controladora o en el teach, porque el

programa no tiene esta funcionalidad, ya que no es posible enviar este comando por software.

- Es aconsejable buscar un proyecto parecido al que se quiere realizar y tratar de contactar con la persona que lo desarrolló, ya que esto permite una gran recolección de información importante para el proyecto, y finalmente es un muy buen referente para comparar resultados e incluir mejores funcionalidades.
- Cuando vayamos a realizar algún movimiento del robot, por sobre el 10% de su velocidad, es siempre recomendable avisar a las personas que se encuentran cerca del robot, que tengan la precaución de alejarse hasta una posición segura, para que nadie salga lastimado y el equipo no se estropee.
- Cuando vayamos a realizar algún movimiento desde el teach, y recibimos constantemente alarmas de la controladora, hay que revisar que estemos presionando bien el botón de seguridad ubicado tras el teach y que las llaves de seguridad estén colocadas en las posiciones correctas.

## 5 ANEXOS

### 5.1 METODOS CLASE CLSRV-2AJ

METODOS			
NOMBRE	DESCRIPCION	PARAMETRO	RETORNA
setPar	Setea el puerto de comunicación	Nombre del puerto COM (string)	Nada
abrirCom	Abre el puerto de comunicación	Ninguno	Nada
cerrarCom	Cierra el puerto de comunicación	Ninguno	Nada
Escribir	Escribe las instrucciones en el formato correcto para la controladora	La instrucción que se desea ejecutar (string)	Nada
Comando	Permite enviar comandos de movimiento	El comando que se desea ejecutar (string)	Nada
leerXYZ	Devuelve las coordenadas X,Y,Z,A,B en las que se encuentra el brazo	Ninguno	Vector de posiciones XYZAB (string [5])
moverXYZ	Permite desplazar el gripper con movimientos lineales	Velocidad (double), X (double), Y (double), Z (double), A (double), B (double)	Nada
moverArticulación	Permite mover cada articulación individualmente	Sentido de giro (int), articulación (int), velocidad (double), ángulo de giro (double)	Nada

resetAlarm	Resete todas las alarmas activadas	Ninguno	Nada
servoOn	Activa los servos	Ninguno	Nada
servoOff	Apaga los servos	Ninguno	Nada
abrirGripper	Abre el gripper	Ninguno	Nada
cerraGripper	Cierra el gripper	Ninguno	Nada
moverPosMem	Desplaza el brazo a la posición guardada en memoria	Posición a la que se quiere mover (string)	Nada
leerProgramas	Lista los programas almacenados en la controladora	Ninguno	Vector de programas (String [n])
posList	Devuelve la lista de posiciones del programa seleccionado	Programa del que se quiere leer las posiciones (string)	Vector de posiciones (string [n])
setVelJoint	Define la velocidad para movimientos cilíndricos	Velocidad de 0% a 100% (double)	Nada
setVelXYZ	Define la velocidad para movimientos lineales	Velocidad en mm/s (double)	Nada
moverAPos	Realiza movimientos cilíndricos con parámetros XYZAB	Velocidad (double), X (double), Y (double), Z (double), A (double), B (double)	Nada
reachPos	Valida si la posición actual de la pinza es la posición deseada	Posición deseada (int [5])y posición actual (int [5])	boolean

SPI es un bus de tres líneas, sobre el cual se transmiten paquetes de información de 8 bits. Cada una de estas tres líneas porta la información entre los diferentes dispositivos conectados al bus. Cada dispositivo conectado al bus puede actuar como transmisor y receptor al mismo tiempo, por lo que este tipo de comunicación serial es full dúplex. Dos de estas líneas transfieren los datos (una en cada dirección) y la tercera línea es la del reloj.

Algunos dispositivos solo pueden ser transmisores y otros solo receptores, generalmente un dispositivo que transmite datos también puede recibir.

El bus SPI emplea un simple registro de desplazamiento para transmitir la información.

Los dispositivos conectados al bus son definidos como maestros y esclavos. Un maestro es aquel que inicia la transferencia de información sobre el bus y genera las señales de reloj y control.

Un esclavo es un dispositivo controlado por el maestro. Cada esclavo es controlado sobre el bus a través de una línea selectora llamada Chip Select o Select Slave, por lo tanto el esclavo es activado solo cuando esta línea es seleccionada. Generalmente una línea de selección es dedicada para cada esclavo.

En un tiempo determinado  $T_1$ , solo podrá existir un maestro sobre el bus. Cualquier dispositivo esclavo que no esté seleccionado, debe deshabilitarse (ponerlo en alta impedancia) a través de la línea selectora (chip Select).

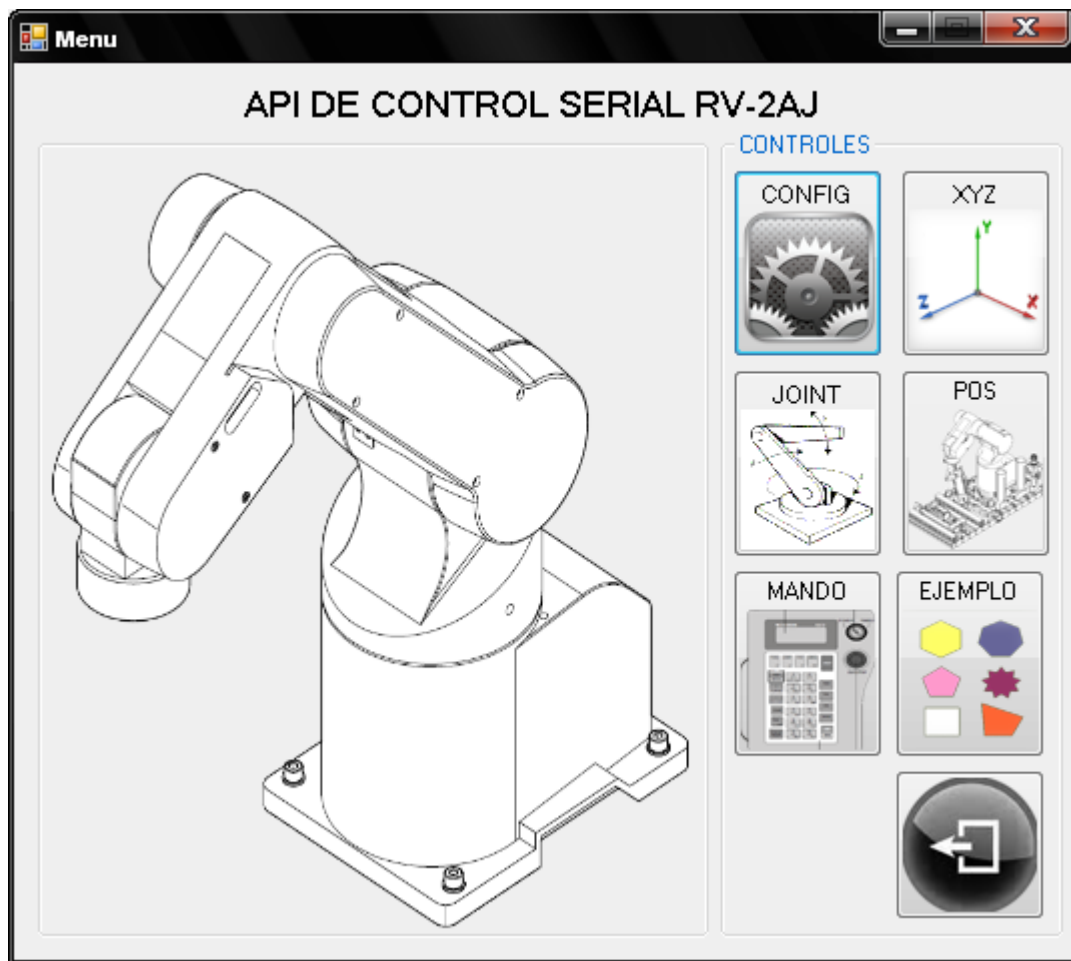
## **5.2 MANUAL DE USUARIO**

Se va a mostrar de manera conceptual la utilización de las pantallas del sistema donde cada botón o representación gráfica será descrita de manera que el usuario se familiarice con el programa.

### **5.2.1 PANTALLA PRINCIPAL DEL PROGRAMA**

Es el menú principal del programa desde donde se puede acceder a todas las funcionalidades del API.

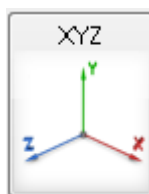




- **Configuraciones**, permite acceder al panel de configuraciones



- **Modo XYZ**, permite acceder al panel de desplazamiento en XYZ



- **Modo Joint**, permite acceder al panel de desplazamiento por articulación.



- **Modo posición**, permite acceder al panel para desplazarse a las posiciones guardadas en la memoria de la controladora.



- **Mando**, permite acceder al panel de control tipo Teach.



- **Ejemplo**, permite acceder al programa de ejemplo.



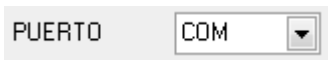
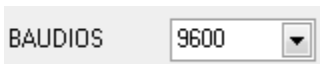
- **Salir**, termina la aplicación y cierra el programa.



### 5.2.2 PANTALLA DE CONFIGURACIÓN

En esta pantalla se definen los parámetros para la comunicación serial.



-   
Muestra los coms que están activos, para que seleccione el puerto de salida.
-   
Velocidad de comunicación serial.

- ▼

Bits de datos a enviar.

- ▼

Tipo de paridad a utilizar.

- ▼

Cuantos bits de parada se van a utilizar.



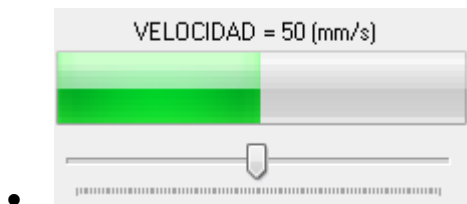
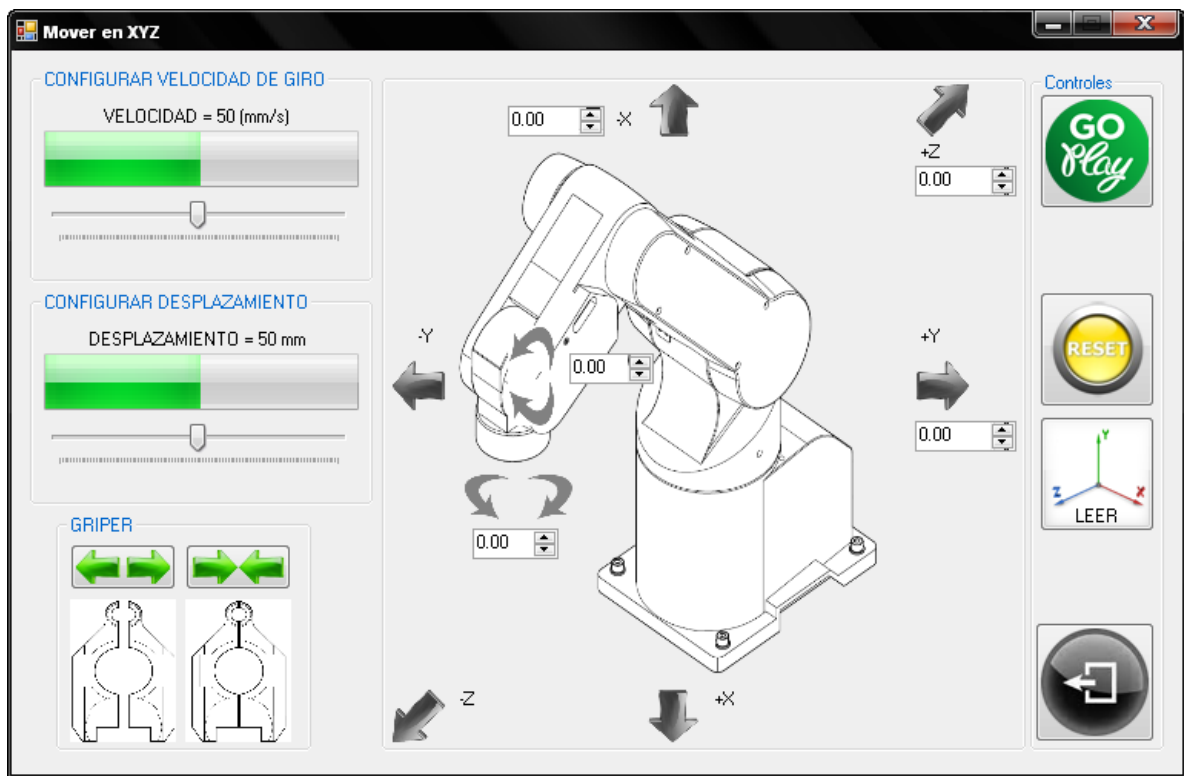
Aceptar las configuraciones y salir del panel de configuraciones.



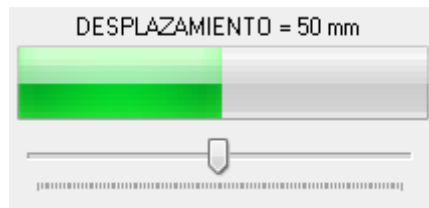
Salir del panel de configuraciones.

### 5.2.3 PANTALLA DE MOVIMIENTO EN MODO XYZ

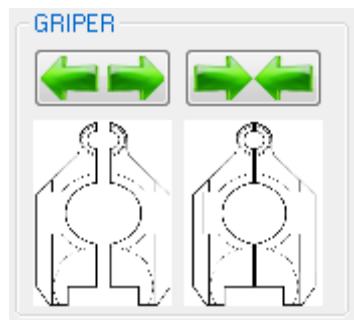
Esta pantalla permite el mover el brazo en los planos XYZ AB, abrir / cerrar el gripper y moverse a una posición específica en XYZ AB.



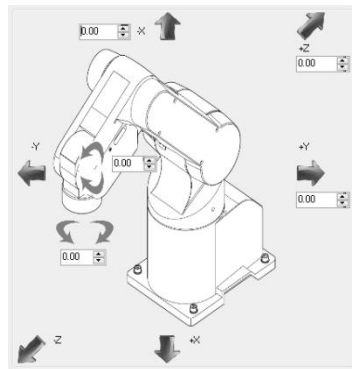
Define la velocidad en mm/s a la que se moverá el robot cuando realice movimientos lineales.



Define la distancia a desplazarse por cada movimiento.



Abre o cierra el gripper.



Permite mover en XYZAB, presionando sobre las flechas. El eje vertical corresponde al X del robot, el eje horizontal al Y del robot y el eje diagonal a Z del robot. La primera articulación de la muñeca es A y la segunda es B.



- Posiciona el robot en las coordenadas deseadas, pero sin hacer movimientos lineales sino cilíndricos.



- Resetea todas las alarmas.



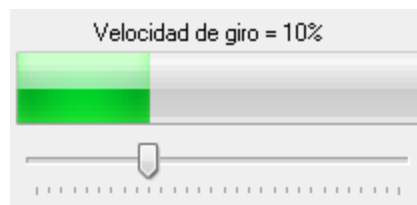
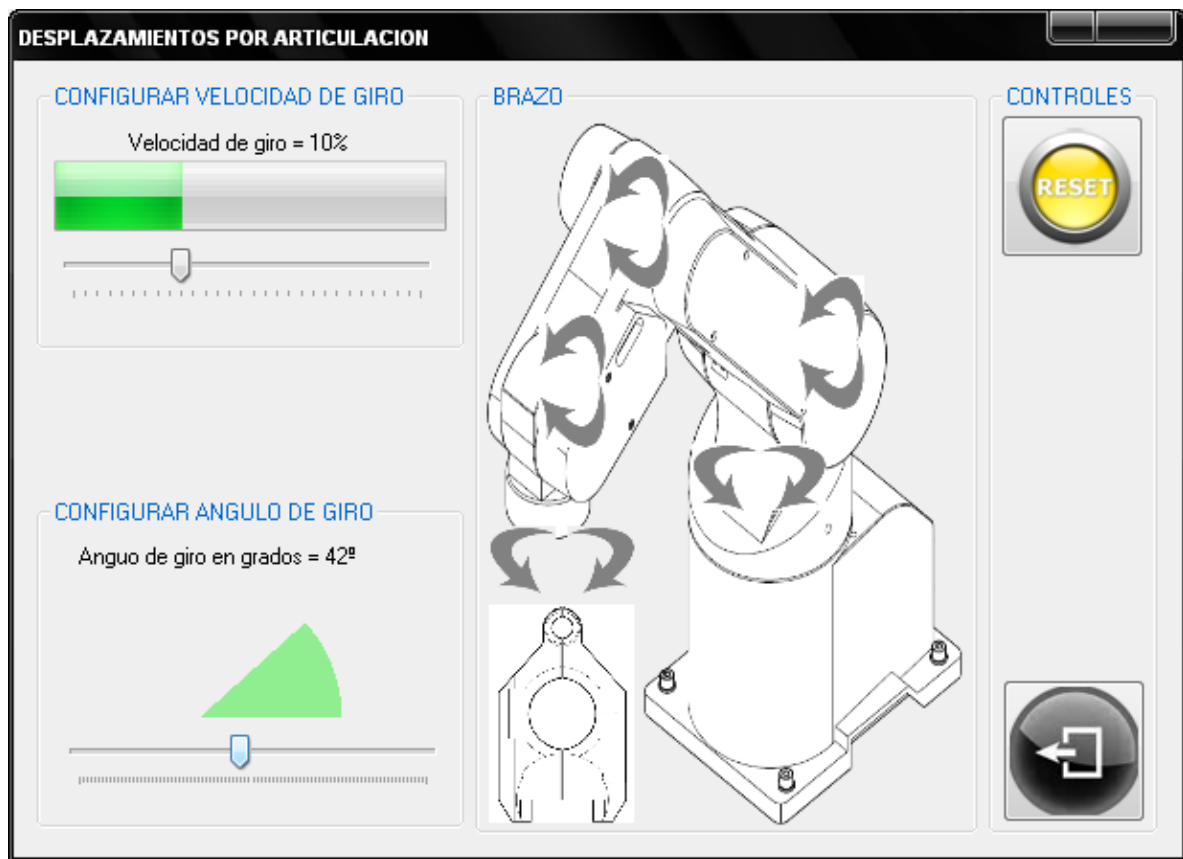
- Obtiene las posiciones XYZAB y actualiza las posiciones al programa.



- Cierra el panel de movimiento en XYZ.

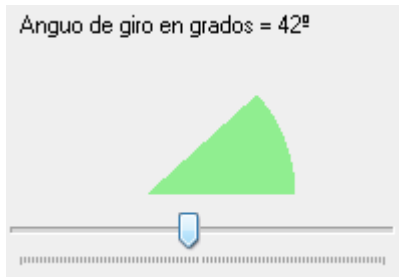
#### 5.2.4 PANTALLA DE MOVIMIENTO POR ARTICULACIÓN

Esta pantalla permite mover el robot por cada una de sus articulaciones y abrir/cerrar el gripper.



- setea la velocidad de giro angular de las articulaciones entre 0% y 100%.





- setea el ángulo a girar por cada una de las articulaciones. Muestra una representación gráfica del ángulo a girar, que está entre  $1^\circ$  y  $90^\circ$ .



- 

Resetea todas las alarmas.

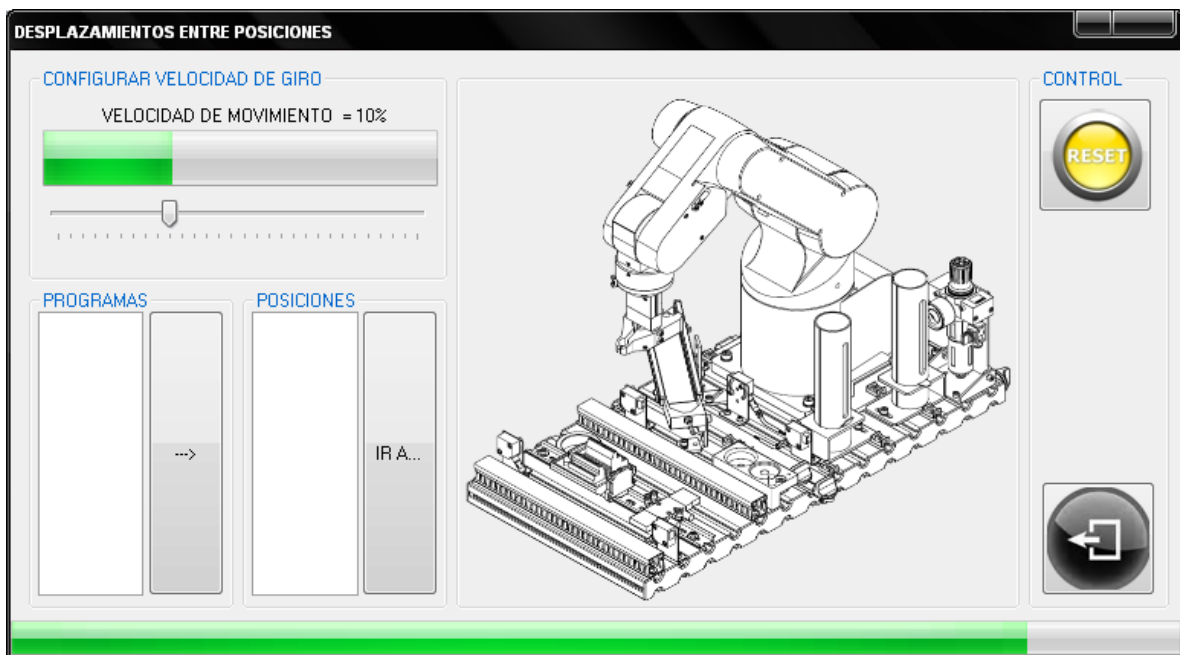


- 

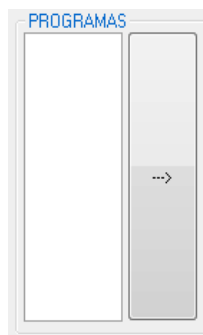
Cierra el panel de movimiento por articulaciones.

### 5.2.5 PANTALLA DE MOVIMIENTO POR POSICIONES

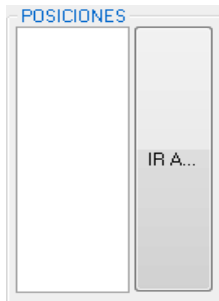
Esta pantalla permite moverse entre posiciones según el programa seleccionado.



Setea la velocidad a la cual se moverá a la posición seleccionada.



Muestra los programas almacenados en memoria y con el botón adquiere las posiciones del programa seleccionado.



- Muestra las posiciones del programa seleccionado y con el botón le ordena al robot moverse.



- Resetea todas las alarmas.



- cierra el panel de movimiento por posiciones.

## 6 BIBLIOGRAFÍA

- BRAUNL, Thomas, Embedded Robotics, 2<sup>da</sup> Edición , Editorial Springer, Australia 2006.
- OLLERO, Aníbal, Robótica, Manipuladores y robots móviles, Editorial Alfaomega y Marcombo, España, 2001.
- BARRIENTOS, Antonio y otros, Fundamentos de la Robótica, Edición 2<sup>da</sup>, Editorial McGraw Hill, España, 2007
- FU y otros, Robótica, control, detección, visión e inteligencia, México, 1990
- WEHNER, Beate, Catalog of ideas, Robots Industriales, Unidades de control, mandos, Augsburg, Alemania, 2010, p151.
- <http://es.wikipedia.org/wiki/Rob%C3%B3tica>
- <http://es.wikipedia.org/wiki/Robot>
- <http://www.superrobotica.com/Tutoriales.htm>
- Manuales de FESTO.
  - 648814\_Manual\_Robot.pdf
  - 132308\_Instructions\_Robot.pdf
  - 132309\_Specifications\_Robot.pdf
  - 133798\_Instructions\_Controller.pdf
  - 133799\_Hardware\_Controller.pdf
  - Teaching\_positions\_RV\_2AJ\_EN.pdf
- Basic serial port listening application
- <http://www.codeproject.com/KB/system/SerialPortListener.aspx>

## REFERENCIAS WEB

- **RV-2AJ** [Web1]  
Da información general sobre el robot RV-2AJ de Mitsubishi.  
<http://es.wikipedia.org/wiki/ASIMO>
- **ROBOT WORX** [Web2]  
Página web del fabricante. Presenta una breve descripción sobre el brazo robótico tipo puma.  
<http://www.robots.com/motoman.php?robot=es280d>
- **MONOGRAFIAS; EQUIPOS NEUMATICOS** [Web3]  
Página en donde se describe la composición de los cilindros neumáticos.  
[http://www.kalipedia.com/kalipediamedia/ingenieria/media/200708/21/informati ca/20070821klpinginf\\_124.les.SCO.jpg](http://www.kalipedia.com/kalipediamedia/ingenieria/media/200708/21/informati ca/20070821klpinginf_124.les.SCO.jpg)
- **FANUC** [Web4]  
Página del fabricante de equipos industriales y muestra descripción de las capacidades técnicas de los motores.  
<http://www.fanuconline.com/images/motors.jpg>
- **MOTORES ELECTRICOS** [Web5]  
En esta página se describe el funcionamiento de diferentes tipos de motores.  
<http://motores-sotorosales.blogspot.com/>
- **MONOGRAFIAS, ROBOTICA Y APLICACIONES** [Web6]  
En esta página se muestra una descripción de la robótica y sus aplicaciones.  
<http://www.monografias.com/trabajos10/robap/robap.shtml>
- **MONOGRAFIAS, ROBOTICA Y APLICACIONES** [Web7]  
En esta página se muestra una descripción de la robótica y sus aplicaciones.  
<http://www.monografias.com/trabajos10/robap/Image3256.gif>
- **MONOGRAFIAS, ROBOTICA Y APLICACIONES** [Web8]  
En esta página se muestra una descripción de la robótica y sus aplicaciones.  
<http://www.monografias.com/trabajos10/robap/robap.shtml>
- **PHOTOMOBWARE, DOCUMENTOS TECNICOS** [Web9]  
Describe el funcionamiento de robots industriales de soldadura.  
[http://www.photomobiware.com/tech/technical32.php?psps\\_page=2](http://www.photomobiware.com/tech/technical32.php?psps_page=2)

- **KALIPEDIA** [WEB10]  
En esta página se encuentra información sobre brazos industriales y sus características.  
[http://www.kalipedia.com/kalipediamedia/ingenieria/media/200708/21/informati-ca/20070821klpinginf\\_124.les.SCO.jpg](http://www.kalipedia.com/kalipediamedia/ingenieria/media/200708/21/informati-ca/20070821klpinginf_124.les.SCO.jpg)
- **ESPOL** [Web 11]  
En esta página se encuentra información sobre brazos industriales y sus características.  
[http://blog.espol.edu.ec/jatacuri/files/2010/11/3006320129\\_09e41cff8a\\_o.jpg](http://blog.espol.edu.ec/jatacuri/files/2010/11/3006320129_09e41cff8a_o.jpg)
- **LOGISMARKE**T [Web 12]  
Página dedicada a la venta de protecciones para robots industriales, que contiene ciertos datos técnicos sobre brazos robóticos para manejo de alimentos  
<http://www.logismarket.es/ip/jke-robotics-fundas-para-proteccion-de-robots-funda-para-robot-de-mecanizado-492382-FGR.jpg>
- **MOTORES SOTO** [Web 13]  
Página de la empresa, dedicada a la comercialización de motores industriales, que contiene información técnica.  
<http://motores-sotorosales.blogspot.com/>
- **IZARO** [Web 14]  
Da información sobre robots para torneear tubos de plástico.  
[http://www.izaro.com/files/contenidos/1305530443\\_1g.jpg](http://www.izaro.com/files/contenidos/1305530443_1g.jpg)
- **ANZEVE** [Web 15]  
Página sobre robots dedicados a aplicaciones de fundición y acería.  
[http://www.anzeve.com/images/brokk/c1\\_aplicacion\\_acerías\\_y\\_fundiciones.jpg](http://www.anzeve.com/images/brokk/c1_aplicacion_acerías_y_fundiciones.jpg)
- **WIKIMEDIA** [Web 16]  
Página con información sobre robots industriales dedicados a automatización de procesos productivos, contiene información técnica y características.  
[http://upload.wikimedia.org/wikipedia/commons/thumb/8/8a/Automation\\_of\\_foundry\\_with\\_robot.jpg/250px-Automation\\_of\\_foundry\\_with\\_robot.jpg](http://upload.wikimedia.org/wikipedia/commons/thumb/8/8a/Automation_of_foundry_with_robot.jpg/250px-Automation_of_foundry_with_robot.jpg)
- **FANUC** [Web17]  
Página de la empresa, describe el funcionamiento y características técnicas del brazo robótico.  
<http://img.robotikka.com/wp-content/uploads/2011/05/fanuc-robotics-robots-delta-robots-genkotsu-los-primeros-robots-de-su-clase-de-6-ejes-623639-FGR.jpg>

- **ESI2** [Web 18]  
Página de desarrollo de proyectos de investigación de automatización industrial.  
<http://www.esi2.us.es/~rubio/proyec.html>
- **ENRYU** [Web 19]  
Contiene información sobre robots para manejo de desechos industriales.  
[http://1.bp.blogspot.com/-KFz\\_pwpnZ3U/TbhdxCyaA\\_I/AAAAAAAAAL10/Eomdyn6XbAE/s1600/ENRYU-T53-header.jpg](http://1.bp.blogspot.com/-KFz_pwpnZ3U/TbhdxCyaA_I/AAAAAAAAAL10/Eomdyn6XbAE/s1600/ENRYU-T53-header.jpg)
- **ECUADOR CIENCIA** [Web 20]  
Página ecuatoriana dedicada a la divulgación de ciencia y tecnología.  
<http://www.ecuadorciencia.org/blog.asp?id=1658>
- **ROBOTICA** [Web 21]:  
En esta página se describen algunos tipos de tecnología industrial, mostrando lo más destacado de cada área.  
<http://robotica.es/category/robots-industriales/>